Market Basket Analysis in Retail

Gerard Reig Grau

May 2017

Advisor: Miquel Sànchez-Marrè
Dept. of Computer Science (UPC)

MASTER IN ARTIFICIAL INTELLIGENCE

FACULTAT D'INFORMÀTICA DE BARCELONA (FIB)
FACULTAT DE MATEMÀTIQUES (FM)
ESCOLA TÈCNICA SUPERIOR D'ENGINYERIA (ETSE)

UNIVERSITAT POLITÈCNICA DE CATALUNYA (UPC) – BarcelonaTech
UNIVERSITAT DE BARCELONA (UB)
UNIVERSITAT ROVIRA I VIRGILI (URV)

Abstract

This Master Thesis memory describes a full end-to-end data science project performed in CleverData, a successful start-up specialized in data mining techniques and analytics tools. This project was performed for one of its clients, which is an important retail company from Spain. The aim of the project was both *the analysis of the possibly different selling behaviour of the stores or shops of the client* and *the analysis of customers' purchase behaviour*, also known as Market Basket Analysis, to confirm the hypotheses from the client regarding the existence of different customer purchasing profiles and different store selling profiles in its company. The project was divided in three tasks.

The first one was oriented to *the study, detection and validation of different behaviour profiles of the shops/stores of the client*. This analysis was done by means of a descriptive process using clustering techniques. In order to guarantee a minimum robustness of the profiles obtained, three clustering algorithms were used: a hierarchical agglomerative clustering technique, a partitional clustering technique with a fixed number of clusters (K-means) and a partitional clustering technique with automatic detection of the number of clusters (G-means). For each algorithm, the output clusters were analysed and compared. First, the similarity of the composition of the clusters between algorithms was analysed. Secondly, the resulting clusters (each partition) from each method were structurally validated using four Clusters Validity Indexes (CVIs): Minimum Cluster Separation Index, Maximum Cluster Diameter Index, Dunn Index and Davies-Bouldin Index. Finally, the best partition was found from a technical point of view.

After that, the client should be able to interpret and validate the meaning of the clusters obtained. Once chosen the partition more meaningful to the client, the second task was devoted to *provide a descriptive analysis of the clusters as meaningful as possible to the client*. To that end, some common techniques tools were used, as the computation of the centroids of the clusters, and the characterisation of each one of the clusters through the variables used. However, an important obstacle appeared in this task. The number of variables was so high (around 400) that made impossible that the client was able to analyse and summarise the selling behaviour profile of the different shops. The proposed solution was to apply a feature selection approach, taking advantage from the clustering process done, and to make an aggregation process of variables with temporal relationship. This way, the information about the cluster to which each store belonged, was recorded as a label of a new created class variable. Then, a Random Forest ensemble technique was selected and applied to the new dataset. This discriminant technique, in addition to be able to predict an unlabelled new instance or observation, provides information about the relevant attributes for the discrimination purpose (i.e., the ones being used in the trees of the forest). Then, based on those most important attributes, the descriptive analysis of each cluster was done, and it could be interpreted and fully understood by the client.

The third task was focused on *the analysis of customers' purchase behaviour* through the analysis of the historic purchase tickets recorded from one year. To identify possible different purchase patterns, it was decided to apply an associative model to find out whether some co-occurrences or associations could be identified. Concretely, the association rules model was used. Because the set of clusters was meaningful to the client, it was decided that the analysis of the purchase behaviour would be done locally to each cluster. Therefore, each cluster was examined to discover associations or co-occurrences of purchase patterns among the customers in each cluster. Hence, some association rules were discovered for the purchase patterns in each store. Two strategies were used to generate the rules: the Lift measure and the Leverage measure.

To summarise and conclude the analysis, a web page was created where the results were published to make easier the access of the client to the results.

Through the memory, it is gradually explained how the project was developed. Since the first step of defining the objectives, until the last results' delivery. In the project, both the Python language and machine learning libraries were used, as well as the BigML tool, which uses machine learning as a service. At the end of the project, the results accomplished were analysed, and the aims of the project were compared against the initial goals of the project, with satisfactory results, both from the client practical point of view, and from a technical point of view.

Contents

Chapter 1

Introduction

1.1 Introduction & Motivation

Retail has evolved through its life. Since the common corner store from the 1900s, until the new e-commerce that has shaken the retail world to its core. This changing process has led to a new era of possibilities for the commerce and the consumer.

Consumers nowadays have a wide range of options. In the past, when the consumer had to buy something, he/she only could choose a product from the catalogue of the store. However, with the new era of information and globalization, the list of options has increased exponentially. Products that some years ago were considered as luxury goods nowadays are considered common, and limitations as geography, season or culture are not more an issue. All of this, lead consumers to have a huge variety of possibilities like new products and new companies. This limitless of possibilities to customers is the one that lead companies start to think new strategies to attract new customers or keep its current customers.

This concept is the one that caused this project. The client is a supermarket chain with a wide list of daily consumers. To increase the experience of the customer and increase its incomes as well, the client decided to invest analysing customer's behaviour purchases using knowledge discovery and data mining process [Novak, 2016], and specifically, was interested in finding item's associations rules within its stores [Association rule, 2017]. This field in retail domain is known as market basket analysis.

Market basket analysis [Kamakura, 2012] encompasses a broad set of analytics techniques aimed at uncovering the associations and connections between specific objects, discovering customer's behaviours and relations between items. In retail, it is used based on the following idea: if a customer buys a certain group of items, is more (or less) likely to buy another group of items. For example, it is known that when a customer buys beer, in most of the cases, buys chips as well. These behaviours produced in the purchases is what the client was interested in. The client was interested in analysing which items are purchased together in order to create new strategies that improved the benefits of the company and customers experience. There are three main issues where market basket analysis is used.

The first one is *the creation of personalized recommendations* [Portugal *et al*., 2015]. This methodology is well known nowadays. During the explosion of the e-commerce,

personalized recommendations has appeared as a part of the marketing process. In a few words, it consists in suggesting items to a customer based on his/her preferences. There are two basic ways to do it. One is suggesting items similar to the ones the customer has purchased in the past (*Content-based approach*), and the other one is looking for similar customers and recommending items that had purchased the similar customers (*Collaborative Filtering approach*). Both strategies are often used for companies in order to realize cross-selling and upselling strategies.

The second one is *the analysis of spatial distribution in chain stores* [R. Kelley & Ming-Long, 2005]. Due the increasing number of products that nowadays exist, physical space in stores has started to be a problem. Increasingly, stores invest money and time trying to find which distribution of items can lead them to obtain more profit. Knowing in advance which items are commonly purchased together, the distribution of the store can be changed to obtain more benefits.

The third one is *the creation of discounts and promotions*. Based in customer's behaviour, special sales can be offered. For example, if the client knows which items are often purchased together, he/she can create new offers for his/her customers.

1.2 Definition of the problem and Objectives

The main aim of the project, according to initial thoughts of the client, was *the detection and analysis of customers' purchase behaviour* (items purchased together). A basic approach could be the creation of a unique rule list for all the tickets and stores. However, this approach lacked efficiency. For instance, suppose the client wants to create a new offer to a specific store based on the rules discovered. It could happen that the daily clients of the store selected do not have the habit to purchase those items, or those items are not even in the store's stock. This could be easily solved using another rule from the set of rules. However, this outlined an important concept: stores could have different behaviours, and this fact originated a second aim of the project: *the analysis of the possibly different selling behaviour of the stores* of the client.

The solution to that problem could be the creation of a store clustering [Pollack, 2016]. Create clusters of stores allow to capture different behaviours. In addition, cluster-local association rules are more realistic and can provide information that is more valuable. The process consisted in create a set of clusters and for each of them, it was selected the store with less distance to the centroid (i.e., the medioid). Then, the association rules of that store were discovered and the results were extrapolated to all the other stores that composed the cluster. With this approach, it was solved the lack of creating a general set of rules for all the stores.

Another issue raised by the client was to define the product level which the association rules should consider. Items belongs to a set of levels. For instance, the item *"patatas lays clasicas*

170 grs" belongs to family *"patatas fritas y fritos"*, the section *"alimentación seca"* and the sector *"alimentación y bebidas"*. The client uses this taxonomy to classify its items for logistics processes [Logistics, 2017]. However, in this project, the client was not interested in finding rules for *items*; it was interested in rules based on *family level*. The client was interested in knowing which product families are purchased together to change its distribution on the stores. In addition, the client was just interested in the items from the sectors: "alimentación y bebidas", "*productos frescos*", "*droguería y perfumeria*" and "*bazar*". Due that, the entire project was done with the items of these sectors.

1.3 Market Basket Analysis strategy

Once analysed and evaluated the client needs, it was defined the approach of the project. The project was divided in three parts.

The first one was the creation of the store clustering. Using the data provided by the client, a dataset was created, where each instance was a store and the features were structural and behavioural information of that store. With the dataset created, three clustering algorithms were used to obtain the clusters, Hierarchical agglomerative, K-means and G-means. To compare the resulting clusters and analyse the quality of them, two experiments were performed. On the one hand, clusters composition was analysed. To do that, for each pair of algorithms, the stores in its clusters were compared. The aim of this composition similarity comparison was to detect whether different algorithms gave similar results. On the other hand, to evaluate the quality of the structure of the clusters. Four Clusters Validity Indexes (CVIs) were used, Minimum Cluster Separation Index, Maximum Cluster Diameter Index, Dunn Index and Davies-Bouldin Index.

The second one was devoted, once selected the proper algorithm and configuration, to the descriptive analysis of the clusters. Using the cluster to which the store belongs, as a new variable in the dataset, a Random Forest ensemble was applied. Then, the most important attributes were used to perform the descriptive analysis and interpretation of the clusters. A characterisation of each cluster through the most relevant variables, and the centroid computation was done.

The third one is the analysis of the historic tickets record from one year. For each cluster, the association rules of the stores in it were discovered according to quality measures: Lift and Leverage.

Finally, it was defined that clusters and association rules had to be retrained periodically. Over time, people behaviour change, and new products or new stores can appear. For that reason, data mining models have to be retrained in order to capture new behaviours.

Chapter 2

State of the art

2.1 A Data Science project

Performing a data science project involves a set of steps to be done. These steps are the skeleton of any data mining/knowledge discovery project. Each one has its own characteristics and objectives, and the sum of all of them, constitute the entire project. The next figure 2.1 is a scheme of the entire process.

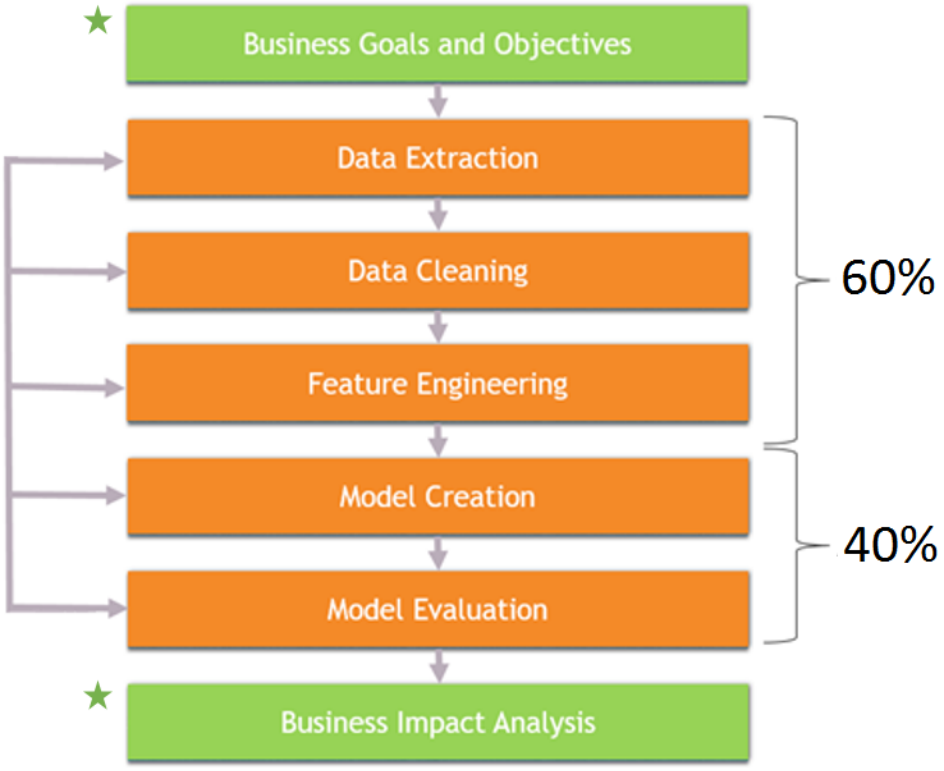


Figure 2.1: A data science project skeleton.

Each rectangle represents a step in the project. On the one hand, the steps from *Data Extraction* until *Model Evaluation* are related to a common data science project. Those steps could have some come back step among them. On the other hand, the steps *Business Goals and Objectives* and *Business Impact Analysis* (star marked) represent the ones where the client has a special impact.

Numbers are time cost approximations of the set of tasks over the total time cost of the project.

Rows are the flux between steps. This is one of the most remarkable characteristics of a data mining project. Flux on traditional projects are sequential, there is just one iteration. However, in this type of problems, the project is developed in several back loops. Therefore, a finished step can be repeated because a new issue or result is obtained.

2.1.1 Business Goals and Objectives

The base of any project is the set of goals and objectives that must be achieved. Decisions and strategies decided in this step, will define the project and the direction where it will be developed.

In this step, the client introduces what he/she expects to achieve using Data Mining techniques. An analysis of the client needs to be performed to understand them, and decide whether they can be achieved using machine learning algorithms or some other technique. If the client needs can be solved using data mining techniques, an approach to the problem is defined and the set of objectives that must be achieved.

2.1.2 Data Extraction

Data extraction is the process of collecting all the available, and presumably interesting, historical data of a company. These data are considered raw because it has not previously received any treatment.

Data extraction is the first step that can be considered part of the data transformation process. Usually this process is a tedious task because companies have data distributed in different sources or databases. In some cases, data is poorly structured or even unstructured. All these aspects convert data extraction process in a hard task.

Nowadays, exist tools prepared to work with this type of problems. Each of them has its own characteristics and methodologies. However, even with this help, the process of collecting data can imply a huge work.

2.1.3 Data Cleaning

Data cleaning is the process of detecting missing values or analysing possible outlier values in the records, and of removing corrupt or inaccurate records from data [Data cleaning, 2017]. Usually, data have errors. These errors can occur being originated from different causes, and the detection of them is vital for the project. Invalid records will imply deterioration of the future model adding noise or false information.

Dara cleaning encompasses the process of removing data which is not relevant or needed as well. Part of the work, is to know which information is relevant or can add value to the algorithms or models used, and treat it for each specific case. Another common situation is that data could be duplicated. Because data emanate from different sources, sometimes the information could be repeated. This provokes an overlap of useless information.

2.1.4 Feature Engineering

Feature engineering is the process of using domain knowledge about the data to create the appropriate features that make machine learning algorithms learn useful patterns from data [Feature engineering, 2017]. This process is fundamental in data mining projects, but it is difficult and expensive. Due this high cost, most of the time of the project is spent in this task. The task consists of finding which features are actually important or needed to add value to the model. This process encompassed the creation and the transformation of features that capture the behaviour and tendencies hidden in the data.

Features used to train a machine learning model affect its performance. As better are the features, better will be the performance. The quality and quantity of features have a huge impact in the model. More than the hyper-parameter configuration of the algorithm, features are the ones that add value to the model. It is worth investing time creating new features, analysing them and transforming data before trying different algorithms.

A complete process of generating an inductive model could be the same as the one used in a cooking recipe. Ingredients would be the dat,a and the algorithm the recipe. If the ingredients are in poor state does not matter that the recipe is the best one of the world, the resulting food will be bad. In the same way, if data has no quality, even with the best algorithm, the results will be bad.

2.1.5 Model Creation

Once obtained the set of features that will be used, the machine learning model is induced /trained from data. Models are feed using the data provided. The dataset, and hence, the learning process can be supervised or unsupervised, and depending on the objective of the problem and the data, supervised or unsupervised machine learning methods are used. A supervised machine learning model uses a supervised dataset, i.e., a dataset which has a special attribute or variable, usually named as the class attribute/variable, which has a label for each observation or instance of the dataset. These labels must be provided by a human expert (here comes the supervised adjective) or obtained by another way. On the contrary, an unsupervised machine learning model uses an unsupervised dataset, i.e., a dataset which has no class attribute/variable at all, and all the observations are unlabelled.

To capture the changing behaviour of the data, machine learning models must be retrained periodically. This period is defined according to the needs of the problem. Alternatively, the machine learning models could be incremental.

2.1.6 Model Evaluation

Model evaluation is the process where the model induced/trained is evaluated using new data. The quality and performance of the model is the result of all the work done through the process. Depending on the type of the model and the type of the target variable, some metrics or others are used to evaluate the quality of the model.

There are two types of evaluation: offline and online. The first one, analyse the performance of a model a priori before deploying it in production. The basic way to do it is with an 80/20 split of the dataset (simple validation) or performing a cross-validation (repeated validation using each fold as test set and the remainder ones as training sets). The second one evaluates the model using real data and analyse its performance.

2.1.7 Business Impact Analysis

The last step in a data mining project is the analysis of the impact that actions had in the problem domain. These actions are executed based on the results obtained through the project.

Companies usually tend to perform projects to obtain a monetary benefit. It can be directly or indirectly. On the one hand, an example of a model used to obtain a direct monetary benefit is one used for churn prediction. It gives a direct income to the company due it prevents to lose clients that would churn. On the other hand, an example of a model used to obtain indirect benefits could be one that group customers based on its behaviours for a posteriori marketing strategy. This model does not feedback with a direct income, but the knowledge of the patterns of those customers can lead to future incomes.

2.2 Data Mining Models

In this project, several data mining models and techniques were used for different tasks. First, descriptive/unsupervised models were used to discover possible groups or clusters of observations (profiles) sharing some interesting features and similar behaviour among themselves. These profiles or clusters should be interpreted by the final users and validated using some structural validation techniques. This structural validation is commonly done in the literature using Cluster Validation Indexes (CVIs).

For the interpretation task of the clusters, some supervised discriminant methods were used to compute the degree of relevance of all the variables. For each cluster, a Random Forest

model, which uses an ensemble of decision trees to make the discrimination process, was used to detect the relevant variables given a concrete cluster.

Finally, once the clusters were interpreted and validated, associative models, and concretely association rules, were used to induce associations among the different variables to get co-occurrence patterns in the data.

Therefore, in the rest of this section, the models and techniques used are described and explained to put them in the adequate context.

2.2.1 Unsupervised/Descriptive Models

There are problems that require discovering the underlying hidden concepts in a dataset or describing the observations/instances by means of obtaining groups or clusters of instances sharing some similarities. Cluster analysis is a Machine Learning task that partitions a dataset and groups together the most similar instances. It separates a set of instances into a number of groups so that instances in the same group, called cluster, are more similar to each other than to those in other groups. Cluster analysis is an unsupervised learning technique. Once the clusters are properly interpreted and validated, it is common to assign a different label to each cluster, creating this way a new qualitative variable in the dataset. Therefore, from then on, the dataset becomes a supervised dataset.

According to the literature [Jain & Dubes, 1988], clustering techniques can be subdivided into partitional clustering techniques and hierarchical clustering techniques by the type of structure imposed on the data. Next, these techniques are described.

2.2.1.1 Partitional Clustering Techniques

A partitional clustering technique generates a single partition of the data in an attempt to recover natural groups present in the data. It tries to obtain a good partition of the observations. The partition is composed by a set of groups or clusters. Thus, this kind of techniques assign each observation to the "best" cluster. This "best" cluster is the one optimizing certain criterion (minimisation of the square sum of distances of the observations to the centroids of the clusters, etc.). Either these algorithms require the number of clusters to be obtained, namely *k*, or some threshold value (classification distance) used to decide whether an observation belongs to a forming cluster or not.

Partitional clustering methods are especially appropriate for the efficient representation and compression of large databases, and when just one partition is needed.

Most popular techniques are the K-means clustering algorithm and the Nearest-Neighbour clustering technique. Next, these algorithms and a variation of K-means algorithm will be described.

2.2.1.1.1 K-means Clustering

One of the most popular partitional clustering algorithms is the K-means clustering algorithm [MacQueen, 1967]. Starting with a randomly initial partition, it explodes the idea of changing the current partition to another one decreasing the sum of squares of distances of the observations to the centroids of the clusters. It converges, possibly to a local minimum, but in general can converge fast in a few iterations. It has a main parameter *k*, which is the number of desired clusters.

The general scheme of the algorithm is as follows:

Algorithm k-means (*k*)
Assign randomly k observations as the centres of the *k* clusters
while any observation changes its cluster membership **do**
    Assigning each observation to its closest cluster centre
    Compute new cluster centres as the centroids of the clusters
endwhile

2.2.1.1.2 G-means Clustering

Sometimes is hard to know in advance how many clusters can be identified in a dataset or simply it is not desired to force the algorithm to output a specific number of clusters. Gaussian-means (G-means) [Hamerly & Elkan, 2003] was designed to solve this issue. G-means use a special technique for running K-means multiple times while adding centroids in a hierarchical way. G-means has the advantage of being relatively resilient to covariance in clusters and has no need to compute a global covariance. The G-means algorithm starts with a small number of k-means centres, and grows the number of centres. Each iteration of the algorithm splits into two those centres whose data appear not to come from a Gaussian distribution. Between each round of splitting, *k*-means is run on the entire dataset and all the centres to refine the current solution.

The test used is based on the Anderson-Darling statistic [Anderson & Darling, 1954]. This one-dimensional test has been shown empirically to be the most powerful normality test that is based on the empirical cumulative distribution function (ECDF).

The general scheme of the algorithm is as follows:

Algorithm G-means (X, α)
Let C be the initial set of centers (usually $C \leftarrow \{\bar{x}\}$, *i.e.*, $k=1$)
repeat
    $C \leftarrow kmeans(C, X)$
    Let $\{x_i|\text{class}(x_i) = j\}$ be the set of datapoints assigned to center c_j
    for each c_j C **do**
        Use a statistical test to detect if $\{x_i|\text{class}(x_i) = j\}$ follow a Gaussian distribution (at conf. level α)
        if the data look Gaussian **then** keep c_j
        else replace c_j with two centers
        endif
    endfor
until no more centers are added

2.2.1.1.3 Nearest-Neighbour Clustering

A natural way to define clusters is by utilizing the property of nearest neighbours; an observation should usually be put in the same cluster as its nearest neighbour. Two observations should be considered similar if they share neighbours.

One of the most used clustering algorithm which is based on the nearest neighbour idea is due to [Lu & Fu, 1978], where the user specifies a threshold, *t*, on the nearest-neighbour distance. If new observations are at a less distance from its nearest neighbour than *t*, then they are assigned to the same cluster than its nearest neighbour.

The general scheme of the algorithm is as follows:

Algorithm Nearest-Neighbour (t)
Let number of clusters (k), k = 1
Assign observation x_1 to cluster C_1
while not all observations are processed **do**
    Find the NN of observation X_i among the observations already assigned to clusters
    Let $d_{i,NNm}$ denote the distance from X_i to its nearest neighbour (NN_m) in cluster m
    if $d_{i,NNm}$ t **then** assign X_i to C_m.
        else set k = k + 1;
            assign X_i to a new cluster C_k
    endif
endwhile

2.2.1.2 Hierarchical Clustering Techniques

A hierarchical clustering process is a nested sequence of partitions. Hierarchical clustering is a general family of clustering algorithms that build nested clusters by merging or splitting them successively. This hierarchy of clusters is represented as a tree, (named as *dendrogram*). A *dendrogram* is a special type of tree structure that provides a convenient picture of a hierarchical clustering. A *dendrogram* consists of a rooted binary tree, where the nodes represent clusters. Lines connecting nodes represent clusters which are nested into one another. Cutting horizontally a *dendrogram* creates a clustering. Figure 2.2 provides a simple example of a dendrogram.

The root of the tree is the unique cluster (conjoint cluster) that gathers all the samples; the leaves being the clusters with only one sample (disjoint clusters). Hierarchical clustering techniques are useful when more than one partition is needed and/or when taxonomies are required like in Medical, Biological or Social Sciences. Anyway, *Dendrograms* are impractical with more than a few hundred observations.

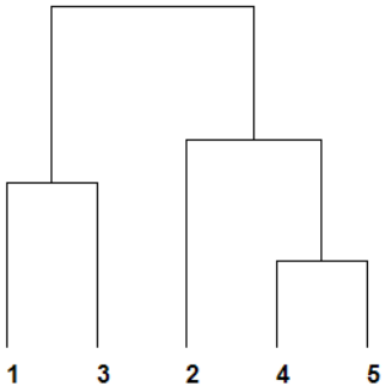


Figure 2.2. A simple example of a *dendrogram* from a hierarchical clustering process

Techniques for hierarchical clustering can be divided into two basic paradigms: agglomerative (bottom-up) and divisive (top-down) approaches. All the agglomerative and some divisive methods (viewed in a bottom-up direction) possess a *monotonicity property*: the dissimilarity between merged clusters is an increasing monotonic function regarding the level of the merger. Therefore, the *dendrogram* can be plotted so that the height of each node is proportional to the value of the intergroup dissimilarity between its two children.

2.2.1.2.1 Agglomerative/Ascendant Techniques

An agglomerative or ascendant hierarchical clustering place each object in its own cluster, and gradually merges these atomic clusters into larger and larger clusters until all objects are in a single cluster. The pair chosen at each step for merging consist of the two clusters with the smallest intergroup dissimilarity (distance). There are several methods implementing this principle. Their difference relies in how they compute the distances (similarities) between the clusters and/or the observations

The general algorithmic scheme for hierarchical agglomerative/ascendant clustering technique [Johnson, 1967] is as follows:

Algorithm Hierarchical agglomerative clustering
Let N be the number of observations to be clustered
Start by assigning each observation to a cluster (N clusters)
Let the distances between the clusters be the distances between the observations they contain
while not all observations are clustered into a single cluster of size N **do**
    Find the closest pair of clusters // differential step of algorithms
    Merge them into a single cluster
    Compute distances between the new cluster and each of the old clusters
endwhile

The different variations of the hierarchical ascendant clustering techniques rely on the step of finding the closest (more similar) pair of clusters. Main used algorithms are known as *single-linkage clustering*, *complete-linkage clustering*, *average-linkage clustering*, *centroid-linkage clustering* and *Ward's method*:

- *Single-linkage clustering* (also called the *connectedness* or *minimum method*) considers the distance between one cluster and another cluster to be equal to the *shortest distance* from any member of one cluster to any member of the other cluster.

- *Complete-linkage clustering* (also called the *diameter* or *maximum method*), considers the distance between one cluster and another cluster to be equal to the *greatest distance* from any member of one cluster to any member of the other cluster.

- *Average-linkage clustering*, considers the distance between one cluster and another cluster to be equal to the *average distance* from any member of one cluster to any member of the other cluster.

- *Centroid-linkage clustering*, considers the distance between one cluster and another cluster to be equal to the *distance* between *the centroids* of each cluster.

- *Ward's method* (also called the *minimum variance method*) [Ward, 1963], which merges in a new cluster (t), the pair of clusters (p, q) minimizing the change in the square-error of the entire clustering $\nabla E_{pq}^2 = e_t^2 - e_p^2 - e_q^2$. The square-error of the entire clustering is the sum of the square-errors for the individual clusters (i.e., sum of squared distances to the centroid for all the observations in a cluster).

The general complexity for agglomerative clustering is $O(n^2 \log(n))$ [Rokach & Maimon, 2005], but for some special cases, optimal efficient agglomerative methods of complexity $O(n^2)$ are known: SLINK [Sibson, 1973] for single-linkage clustering and CLINK [Defays, 1977] for complete-linkage clustering.

2.2.1.2.2 Divisive/Descendent Techniques

Divisive or descendent hierarchical clustering reverses the process by starting with all objects in one cluster and recursively divide one of the existing clusters into two daughter clusters at each iteration in a top-down procedure. The split is chosen to produce two new clusters with the largest intergroup dissimilarity (distance).

In the general case, divisive clustering techniques have a complexity of $O(2^{n-1})$ [Everitt, 2011]. For that reason, divisive methods are not very popular, and this approach has not been studied as extensively as agglomerative methods in the clustering literature. The existent algorithms propose some *heuristic* in order not to generate all possible splitting combinations.

One of the first divisive algorithm in the literature was proposed in [Macnaughton-Smith *et al*., 1965]. It begins by placing all observations in a single cluster G. It then chooses that observation whose average dissimilarity from all the other observations is largest. This observation forms the first member of a second cluster H. Then, it moves to the new cluster H the observations in G whose average distance from those in G is greater than the average distance to the ones in the new cluster H. The result is a split of the original cluster into two children clusters, the observations transferred to H, and those remaining in G. These two clusters represent the second level of the hierarchy. Each successive level is produced by applying this splitting procedure to one of the clusters at the previous level.

Other divisive clustering algorithm was published as the DIANA (DIvisive ANAlysis Clustering) algorithm [Kaufman & Roussew, 1990]. DIANA follows the same strategy proposed by Macnaughton-Smith, but chooses the cluster with the largest diameter (i.e., the one maximizing the distance among its member observations). A possible alternative could be to choose the one with the largest average dissimilarity among its member observations.
An obvious alternate choice is k-means clustering with k = 2, [Steinbach et al., 2000] but any other clustering algorithm producing at least two clusters can be used, provided that the splitting sequence possesses the *monotonicity property* required for a *dendrogram* representation.

2.2.1.3 Clustering Validation Techniques

Once a clustering technique has been applied, the resulting set of clusters must be validated to ensure that the clusters are structurally well formed, and to get the underlying meaning of the clusters. Usually, the real partition of the data is unknown and, therefore, the results from a clustering process cannot be compared with a reference partition by computing misclassification indexes, as in the case of supervised learning.

2.2.1.3.1 Structural Validation of Clusters

Cluster structural validation in clustering field is an open problem. In the literature, most of used techniques for evaluating the clustering results are based on numerical indexes, which

evaluate the validity of the resulting partition from different points of view, known as Cluster Validity Indexes (CVI). A wide number of CVIs can be found in literature and some surveys comparing several CVIs [Halkidi *et al*., 2001]. However, there are currently no clear guidelines for deciding which is the most suitable index for a given dataset [Brun *et al*., 2007]. In fact, there is not an agreement among those indexes, but it seems clear that each one can give some information about a different property of the partition like homogeneity, compactness of classes, variability, etc. All these CVIs refer to structural properties of the partition, which are context-independent, and the evaluation based on them is mainly made in terms of the cluster' topology.

Most common CVIs in the literature are:

- Entropy index
- Maximum Cluster Diameter index (Δ)
- Widest Gap index (*wg*)
- Average Within-Cluster Distance index (*W*)
- Within Cluster Sum of squares index (*WSS*)
- Average Between-Cluster Distance index (*B*)
- Minimum Cluster Separation index(δ)
- Separation index (*Sindex*)
- Dunn index (*D*)
- Dunn-like index
- Calinksi-Harabasz index (*CH*)
- Normalized Hubert Gamma Coefficient ($\Gamma\hat{}$)
- Silhouettes index
- Baker and Hubert index (*BH*)
- Within Between Ratio index (*WBR*)
- C-index
- Davies-Bouldin index (*DB*)

In a recent work [Sevilla-Villanueva *et al*., 2016], it was outlined that indexes evaluate a reduced set of characteristics of a partition. Thus, all indexes can be grouped around 4 basic concepts:

- Indexes measuring compactness of clusters: *Diameter* (Δ), *wg*, *W*, *WSS.*
- Indexes measuring separation between clusters: *B*, *Separation* (δ), *Sindex.*
- Indexes measuring relationships between compactness and separation: *CH*, *Silhouettes*, *Γˆ*, *BH*, *WBR*, *C-Index, DB, and also D, Dunn-like.*
- Indexes measuring chaos in the clusters: Entropy.

Therefore, it would be a good strategy to select one index from a different family to evaluate different properties of the clustering result. In the project, the following four indexes were selected to be used:

Maximum Cluster Diameter (Δ) [Hennig and Liao, 2010] is the maximum distance between any two points that belongs to the same cluster.

$$\Delta = \max_{C_i \in P} \Delta_{C_i}$$
$$\Delta_{C_i} = \max_{o_1, o_2 \in C_i} d(o_1, o_2)$$

Minimum Cluster Separation (δ) is the minimum distance between any two objects that do not belong to the same cluster. In other words, it is defined by the lower separation among all the clusters.

$$\partial = \min_{C_1, C_2 \in P} \partial_{C_1, C_2}$$

$$\partial_{C_1, C_2} = \min_{o_1 \in C_1, o_2 \in C_2} d(o_1, o_2)$$

Dunn Index (*D*) is a cluster validity index for crisp clustering proposed in [Dunn, 1974]. It attempts to identify "compact and well separated clusters"

$$D = \frac{\partial}{\Delta} = \frac{\min_{C_1, C_2 \in P} \partial_{C_1, C_2}}{\max_{C_i \in P} \Delta_{C_i}}$$

Davies-Bouldin Index (*DB*) [Davies & Bouldin, 1979] is a cluster relation of compactness and separation measure. The overall index is defined as the average of indexes computed from each individual cluster. An individual cluster index is taken as the maximum pairwise comparison involving the cluster and the other clusters in the solution.

$$DB = \frac{1}{m} \sum_{C \in P} \max_{C' \in P, C' \neq C} \left(\frac{s_{p_C} + s_{p_{C'}}}{d_p(C, C')} \right)$$

Where $d_p(C, C') = \sqrt[p]{\sum_{k=1}^{K} |\overline{X_{C_k}} - \overline{X_{C'_k}}|^p}$ and $s_{p_C} = \sqrt[p]{\frac{\sum_{o_i \in C} d_p(o_i, o_{ic})^p}{n_C}}$

Where, o_{ic} is the barycenter of the cluster *C* defined as $o_{ic} = (\overline{x_{C_1}}, \ldots, \overline{x_{C_k}})$

2.2.1.3.2 Expert Validation of Clusters

In addition to the structural validation of the clusters, it is very important to make a qualitative validation of the clusters. Usually, the experts make this kind of validation. This validation process consists to carefully look at the composition of the obtained clusters, analyse them, and try to get an interpretation of each one of the clusters.

This interpretation process can be done through some *data summarisation* techniques, like the *computation of the cluster centroids*. A cluster centroid is a prototype showing the most frequent characteristics of the observations belonging to that cluster. This information is very important to illustrate how is the general profile of the observations belonging to a cluster. A centroid is a virtual observation, which is the geometrical centre of the set of observations. It has the same number and type of components than the observations, and usually has the average value of the numerical variables, the mode of qualitative variables, etc. IT provides a very useful information of the prototypical kind of observations of a cluster (low values of variable X1, high values of variable X2, etc.)

In addition, several graphical visualizations of data can help to the interpretation of the clusters (histograms, tables, bivariate plots, letter plots, etc.). All these graphics can help to identify the characteristics of each one of the clusters.

2.2.2 Supervised Discriminant Models

Another common problem in Machine Learning is to obtain a discriminant model from a supervised dataset. Discriminant models are able to discriminate or predict the class label of a new unlabelled instance. Discriminant models are also called classifier models or systems in the literature.

There are several kinds of discriminant methods like Support Vector Machines, Decision Trees, Classification Rules, Bayesian discriminant methods, Case-Based Classifiers, etc. In addition, in the literature there is the approach of working with an ensemble of discriminant methods. As in this project work, Decision Trees, and some ensemble of classifiers approach, concretely Random Forests were used. All these techniques are detailed a bit in the next subsections.

2.2.2.1 Decision Trees

A decision tree is a hierarchical structure (a tree), which can model the decision process of deciding to which class belongs a new example of a concrete domain. In a decision tree, the internal nodes represent qualitative attributes, or discretized numerical ones in some approaches. For each possible value of the qualitative attribute, there is a branch. The leaves of the tree have the qualitative prediction of the attribute that acts as a class label.

Decision Trees has some advantages over other discriminant models. The final model, the tree, is easily interpretable by an expert or end user to understand the decision process, which ends assigning a label to a new unlabelled instance. Another interesting point is that at the same time that the decision tree is constructed, the attributes that have not been used in the building of the tree, are not necessary for a discrimination process. This fact probably means that those unused attributes are not very important. This way, using a decision tree has the

benefit of performing an internal *feature selection* process as an integral part of the procedure. They can manage the presence of irrelevant predictor attributes.

There are different techniques to induce a decision tree from a supervised training dataset. All methods use a top-down recursive procedure with a greedy strategy to select the adequate attribute at each node. The strategy tries to select the most discriminant attribute at each step. The discrimination among the different classes is maximized, when the level of separation or skew among the different classes in a given node is maximized.

The difference among the methods relies on how to estimate which is the most discriminant one. Most common methods in the literature are:

- ID3 method [Quinlan, 1983; Quinlan, 1986]
- CART method [Breiman *et al*., 1984]
- C4.5 method [Quinlan, 1993]

The measures used to compare different decision trees are: the compactness of the tree, the predictive accuracy of the tree, the generalization ability of the tree (scalability). In addition, some approaches propose pruning techniques to reduce the size of the tree and try avoiding overfitting problems.

2.2.2.1.1 Information Gain Methods

One of the most well-known method for inducing a decision tree is the ID3 [Quinlan, 1983; Quinlan 1986] method. On each iteration of the algorithm, it selects the best attribute according to the *Information Gain* criteria.

The *Information Gain* criteria is based on the concept of *Entropy* from information theory. The criteria selects the attribute, which maximizes the information gain. Thus, the ID3 algorithm needs to assess the information gain provided by the use of each one of the considered attributes. The Entropy function measures the ability of each attribute to split the instances in the possible values of the attribute in the best pure (discriminant) form. Purity means that if all instances having the same value for the attribute belongs to the same label is a better attribute than others that are mixing several instances belonging to different labels.

$$Gain(X, A) = Info(X) - Info(X, A)$$

Where X is the set of all instances to be discriminated at each node, and k is the number of different labels of the class attribute.

$$Info(X) = H(X) = -\sum_{i=1}^{k} p_{x \in C_i} * \log_2 p_{x \in C_i}$$

is the entropy at the node before splitting, and

$$Info(X,A) = \sum_{j=1}^{v} p_{x \in Value_j(A)} * Info(\{x | x \in Value_j(A)\})$$

is the amount of information needed to arrive to a perfect classification using the corresponding attribute *A*.

The value of the entropy lies between 0 and log(*k*). The value is log(*k*), when the instances are perfectly balanced among the different classes. This corresponds to the scenario with maximum entropy. The smaller the entropy, the greater the separation in the data.

It selects the attribute which has the smallest entropy (or largest information gain) value. The set *X* is then split by the selected attribute to produce subsets of the data. The algorithm recursively continues on each subset, considering only attributes never selected before.

This Information Gain measure is biased to select attributes with large number of possible values. In order to overcome this bias, Quinlan [Quinlan, 1993] proposed the C4.5 method which uses an extension to information gain known as *gain ratio*. It applies a kind of normalization to information gain using a *split information value*. The *split information value* represents the potential information generated by splitting the training data set *X* into *v* partitions, corresponding to the *v* possible values of the attribute *A*.

$$SplitInfo_A(X) = -\sum_{j=1}^{v} p_{x \in Value_j(A)} * \log_2(p_{x \in Value_j(A)})$$

The GainRatio is defined as follows:

$$GainRatio(X,A) = \frac{Gain(X,A)}{SplitInfo_A(X)}$$

At each node of the tree, C4.5 chooses the attribute of the data that most effectively splits its set of samples into subsets according to the Normalized Information Gain or Gain Ratio. The attribute with the highest Gain Ratio is chosen.

The C4.5 method has been implemented in the J4.8 method in the software WEKA. In next evolutions of the method, Quinlan proposed the C5.0 method, where the most significant feature unique to C5.0 is a scheme for deriving rule sets.

2.2.2.1.2 Impurity Measure Method

Another well-known method is the CART method (Classification And Regression Trees) [Breiman *et al.*, 1984], which base Impurity measure. For example, if p_1 ... p_k is the fraction of the instances belonging to the *k* different classes in a node N, then the Gini-index of impurity Gini(X) of the current node is defined as follows:

$$Gini(X) = 1 - \sum_{i=1}^{k} p_{x \in C_i}$$

Where X is the set of all instances to be discriminated at each node, and k is the number of different labels of the class attribute.

The value of Gini(X) lies between 0 and $1 - 1/k$. The smaller the value of Gini(X), the greater the separation. In the cases where the classes are evenly balanced, the value is $1 - 1/k$.
The Gini Index considers only *a binary split* for each attribute A, say X_1 and X_2. The Gini index of X given that partitioning is a weighted sum of the impurity of each partition:

$$Gini(X, A) = \frac{|X_1|}{|X|} * Gini(X_1) + \frac{|X_2|}{|X|} * Gini(X_2)$$

Finally, the attribute that maximizes the reduction in impurity is chosen as the splitting attribute.

$$\Delta Gini(A) = Gini(X) - Gini(X, A)$$

2.2.2.2 Ensemble Methods

In the literature, there are several works proposing the use of a set of discriminant/classifier models. The aim is to build a discriminant/classifier model by combining the strengths of a collection of simpler base models. There are several ways of implementing this idea. Some approaches are based on resampling the training set, others on using different discriminant/classifier methods, others on varying some parameters of the classifier methods, etc. Finally, the ensemble of methods is used to combine the output of each classifier, i.e., the predicted label, by means of a (weighted) majority voting.

In next subsections, the most common approaches are detailed: bagging, boosting and random forests.

2.2.2.2.1 Bagging

The *Bagging* (Bootstrap Aggregating) strategy [Breiman, 1996] propose to create ensembles by repeatedly and randomly resampling the training data. Given a training set of size *n*, create *m* samples of size *n* by drawing *n* examples from the original data, with replacement. These are referred to as *bootstrap samples*. Each bootstrap sample will contain different training examples, and the rest are replicates. For each sample, the classifier method is used to induce one model. At the testing step, all models are used, and their output labels are combined in a majority vote scheme.

This approach has often been shown to provide better results than single models in certain scenarios. This approach can reduce the variance of classifiers improving the accuracy, because of the specific random aspects of the training data. Decreases error by decreasing the variance in the results due to unstable learners (like decision trees) whose output can change dramatically when the training data is slightly changed.

2.2.2.2.2 Boosting

Boosting [Freund, 1995] is a common technique used in classification. The idea is to focus on successively difficult instances of the data set, to create models that can classify these instances more accurately, and then use the ensemble scores over all the components. A holdout approach is used to determine the incorrectly classified instances of the data set. Thus, the idea is to sequentially determine better classifiers for more difficult instances, and then combine the results to obtain a meta-classifier, which works well on all the dataset.

To focus on difficult instances, they are given weights. At each iteration, a new hypothesis is learned and the examples are reweighted to focus the system on examples that the most recently learned classifier got wrong.

General boosting algorithm can be expressed as follows:

Algorithm Boosting
Set all examples to have equal uniform weights
for *t* from 1 to T **do**
    Learn a hypothesis, h_t from the weighted examples
    Decrease the weights of examples h_t classifies correctly
endfor

Base (weak) learner must focus on correctly classifying the most highly weighted examples while strongly avoiding over-fitting. During testing, each of the T hypotheses get a weighted vote proportional to their accuracy on the training data.

One of the most used boosting approach is the AdaBoost (Adaptive Boosting) algorithm [Freund & Shapire, 1997], for building ensembles, that empirically improves generalization performance.

2.2.2.2.1 Random Forests

Random forests [Breiman, 2001] is a method that proposes to use sets of unpruned decision trees aiming to reduce the error of the single classifiers. Each decision tree is built splitting at each node using a random selection of features and the training data for each tree is a bootstrap sample of the training data.

A number *m* is specified much smaller than the total number of attributes *M* (e.g., *m = sqrt* (*M*) or *m = int* (log2 *M* +1)). At each node, *m* attributes are selected at random out of the *M*. The split used is the best split, according to the criteria used (information gain, gain ratio, Gini index of impurity, etc.), on these *m* attributes.

At the testing step of unclassified instances, final classification is done by majority vote across the trees. Usually, error rates compare favourably to AdaBoost. It is more robust with respect to noise, and efficient on large data.

Random forests are closely related to bagging, and in fact bagging with decision trees can be considered a special case of random forests, in terms of how the sample is selected (bootstrapping). In addition, they provide an estimation of the importance of features in determining classification.

2.2.3 Associative Models

In the same way that descriptive models try to find relationships among the instances of a database, there are associative models, which aim to find some relationship among the variables in a dataset. There are problems that require finding meaningful relationships among variables in large datasets across thousands of values, e.g., discovering which products are buy together by customers (i.e., *market basket analysis*), finding interesting web usage patterns, or detecting software intrusion. These problems can be solved using Associative models. Among the associative models, most commonly used methods are Association Rules techniques, Qualitative Reasoning models, and other statistical methods like Principal Component Analysis (PCA), etc. Association Rule techniques have been used in this work for its easiness of interpretation by the experts. For that reason, they are described in the next subsection.

2.2.3.1 Association Rules

The main goal of the Association Rules technique is to obtain a set of association rules which express the correlation among attributes, from a database of item transactions. These techniques were originated in the field of Knowledge Discovery in large databases. Thus, accordingly, the common terminology talks about transactions, databases, and items in the transactions, because these techniques were first applied to the market basket analysis domain, and the *transactions* were composed of the different *items* bought by a customer.

Given a database consisting of a set of *transactions* $D = \{t_1, t_2, \dots, t_n\}$, and given $I=\{i_1, ..., i_n\}$ be a set of *n* attributes called *items*.
Each transaction in *D* has a unique transaction ID and contains a subset of the items in *I*:

t_1: i_2, i_3, i_4, i_6, i_9
t_2: $i_1, i_2, i_4, i_7, i_8, i_9$

t_3: i_2, i_4, i_5, i_6
t_4: i_1, i_3, i_4, i_8, i_9, i_{10}
. . .
t_n: i_3, i_4, i_6, i_9

The issue is to obtain common *patterns of co-occurrence* of the same *items* along the database. Of course, in order that the co-occurrences found in the database have some interest, the database should have enough number of transactions in order that the co-occurrence appear a sufficient number of times. This minimum number of times required for a co-occurrence is named as the *minimum support* (*minsup*) of the rule expressing the co-occurrence.

For instance, the following *common patterns* can be obtained from the previous database:

i_2, i_4
i_4, i_9
i_2, i_4, i_9
i_3, i_4, i_9
i_3, i_4, i_6, i_9
.....

From a *common pattern*, several *association rules* can be generated. An *association rule* is defined as an implication of the form:

$$X \Rightarrow Y$$
$$\text{where } X, Y \subseteq I \text{ and } X \cap Y = \varnothing$$

Every rule is composed by two different *sets of items*, also known as *itemsets*, X and Y. X is called *the antecedent* or left-hand-side (LHS) of the rule and Y is called *the consequent* or right-hand-side (RHS) of the rule.

For instance:

$i_2 \Rightarrow i_4$	$i_4 \Rightarrow i_2$		
$i_4 \Rightarrow i_9$	$i_9 \Rightarrow i_4$		
$i_2 \Rightarrow i_4 \wedge i_9$	$i_4 \Rightarrow i_2 \wedge i_9$	$i_9 \Rightarrow i_2 \wedge i_4$	
$i_2 \wedge i_4 \Rightarrow i_9$	$i_2 \wedge i_9 \Rightarrow i_4$	$i_4 \wedge i_9 \Rightarrow i_2$	
$i_3 \wedge i_4 \Rightarrow i_9$	$i_3 \wedge i_9 \Rightarrow i_4$	$i_4 \wedge i_9 \Rightarrow i_3$	
$i_3 \wedge i_4 \wedge i_6 \Rightarrow i_9$	$i_3 \wedge i_4 \wedge i_9 \Rightarrow i_6$	$i_3 \wedge i_6 \wedge i_9 \Rightarrow i_4$	
$i_3 \wedge i_4 \wedge i_6 \Rightarrow i_9$	$i_3 \wedge i_4 \Rightarrow i_6 \wedge i_9$	$i_3 \wedge i_6 \Rightarrow i_4 \wedge i_9$	. . .
. . .			

In the general case of application of association rules, an *item* is an attribute-value pair, and the term *itemset* is the combination of items that have a minimum specified support *(minsup)*. Next, the main concepts related to association rules are defined:

- *Support of an itemset* [Agrawal *et al*., 1993]
  The support value of X with respect to T is defined as the number of transactions

(instances) in the database, which contains the itemset X.

$$supp(X) = |\{t \in T | X \subseteq t\}| \quad \text{(absolute definition)}$$

$$supp(X) = \frac{|\{t \in T | X \subseteq t\}|}{|T|} \text{ (relative definition)}$$

- *Support of a rule* [Agrawal *et al*., 1993]
  The support value of a rule, X ⇒ Y, with respect to T is defined as the percentage of all transactions (instances) in the database, which contains the itemset X and the itemset Y.

$$supp(X \Rightarrow Y) = \frac{supp(X \cup Y)}{|T|}$$

- *Coverage of a rule*
  Coverage is sometimes called antecedent support or LHS support. It measures how often a rule, X ⇒Y, is applicable in a database.

$$coverage(X \Rightarrow Y) = supp(X)$$

- *Confidence*/*Strength of a rule* [Agrawal *et al*., 1993]
  The confidence value of a rule, X ⇒ Y, with respect to a set of transactions T, is the proportion of the transactions that contains X, which also contains Y.

$$conf(X \Rightarrow Y) = \frac{supp(X \cup Y)}{supp(X)}$$

- *Leverage/Piatetsky-Shapiro Measure (PS) of a rule* [Piatetsky-Shapiro, 1991]
  Leverage value of a rule, X ⇒ Y, measures the difference between the probability of the rule and the expected probability if the items were statistically independent.

$$leverage(X \Rightarrow Y) = supp(X \Rightarrow Y) - supp(X) * supp(Y)$$

  It ranges from [-1, +1] indicating 0 the independence condition.

- *Lift/Interest of a rule* [Brin *et al*., 1997]
  Lift value of a rule, X ⇒ Y, measures how many times more often X and Y occur together than expected if they were statistically independent.

$$lift(X \Rightarrow Y) = \frac{supp(X \cup Y)}{supp(X) * supp(Y)} = \frac{conf(X \Rightarrow Y)}{supp(Y)}$$

  It ranges from [0, +∞] where a lift value of 1 indicates independence between X and Y, and higher values indicates a co-occurrence pattern.

The different methods to induce the association rules are interested in rules with a *minimum support* (*minsup*) to outline a repetitive co-occurrence pattern, and with *high confidence*, meaning that the rules are highly accurate (both antecedent and consequent of the rule are satisfied). Also, high values of *lift* are desirable to indicate a co-occurrence pattern strength.

Most well-known methods in the literature are the following:

- Apriori algorithm [Agrawal & Srikant, 1994] was one of the earliest association rules method. In fact, the Apriori algorithm computes just the large itemsets which their support is higher than the minimum support (*minsup*) threshold. It uses a breadth-first search strategy to generate the itemsets: starting from large 1-itemsets, it computes afterwards large 2-itemsets, then large 3-itemsets and so on until the maximum number of attributes available. It uses a candidate generation function, which filters impossible large k-itemsets candidates, because they have subsets of large k-1-itemsets, which do not have a minimum support.

  After the Aprori algorithm, the candidate rules must be generated trying all the possible combinations of the items in the antecedent or the consequent of the rule. The rules are filtered, and just only the ones with a confidence value higher than the minimum confidence bound are shown.

- Eclat (Equivalence CLAss Transformation) [Zaki, 2000; Zaki *et al*., 1997] is a depth-first search algorithm using set intersection. It uses a *vertical tid-list* database format where it associates with each itemset, a list of transactions in which it occurs. All frequent itemsets can be enumerated via simple tid-list intersections. In addition, a lattice-theoretic approach to decompose the original search space (lattice) into smaller pieces (sublattices) which can be processed independently in main-memory is used. Eclat uses a prefix-based equivalence relation for the decomposition of the lattice and a bottom-up strategy for enumerating the frequent itemsets within each sublattice. Eclat requires only a few database scans, minimizing the I/O costs, and it is suitable for both sequential as well as parallel execution with locality-enhancing properties.

  The association rules are generated after the Eclat method, using the same procedure as Aprori and other methods.

- FP-growth (Frequent Pattern growth) [Han *et al*., 2004; Han *et al*., 2000] proposed a novel frequent-pattern tree structure (FP-tree), which is an extended prefix-tree structure for storing compressed, crucial information about frequent patterns, and develop an efficient FP-tree based mining method, FP-growth, for mining the complete set of frequent patterns by pattern fragment growth. Efficiency is achieved with a large database, which is compressed into a condensed, smaller data structure, FP-tree which avoids costly, repeated database scans. The FP-tree-based mining adopts a pattern-fragment growth method to avoid the costly generation of a large number of candidate sets. Moreover, a partitioning-based, divide-and-conquer method is used to decompose the mining task into a set of smaller tasks for mining confined

patterns in conditional databases, which dramatically reduces the search space.

- Filtered-top-k Association discovery [Webb, 2011] is an association technique that focuses on finding the most useful associations for the user's specific application. It tries to overcome the problem of not finding relatively infrequent associations (lower support) but even very interesting associations that most frequent association mining paradigm would not discover. The user specifies three parameters: a measure of how potentially interesting an association is, filters for discarding inappropriate associations, and the number of associations to be discovered, k.

  Any of the numerous measures of an association's worth existing in the literature (lift, leverage, etc.) may be used. Filters can be imposed such as a requirement that associations be non-redundant [Bastide *et al*., 2000; Zaki, 2004], productive [Webb 2006] or pass statistical evaluation [Webb, 2007]. The system finds the k associations that optimise the specified measure within the constraints of the user-specified filters. This solves directly the problems of controlling the number of associations discovered and of focusing the results on associations that are likely to be interesting. It is often possible to derive very efficient search by using k together with the objective function and filters to constrain the search [Hämäläinen, 2010; Pietracaprina *et al*., 2010]. The result is that association mining can be performed efficiently, focusing on associations that are likely to be interesting to the user, without any need for a minimum support constraint.

2.3 BigML Tool

One of the most discussed topics in the Big Data and Machine Learning projects are the methods and tools used. Searching on the internet, reading articles, or speaking with other companies can provide a huge variety of options. Each method or tool has its own properties and advantages; however, the study of them is a tedious task.

The variety of resources is a double-edged sword. Whenever a project start, one can get lost over this huge amount of options. Spending time thinking which tool use can imply to reduce effort and time on the future. A bad selection of tools can consequence into future problems.

The main tool used in this project for the algorithms was BigML [BigML, 2017]. BigML is a pioneer system of machine learning as a service. Is a highly scalable, cloud based machine learning service that is easy to use, seamless to integrate and instantly actionable.

What makes BigML special is that is a simple, visual and powerful tool that makes data mining projects more flexible. As Francisco J. Martín, Co-Founder and CEO of BigML said

in an interview: "*Es una herramienta que sirve para aprender de los datos de forma muy fácil".*[1]

The service offers a wide range of different supervised and unsupervised algorithms. Moreover, it has resources that allows the user to create workflows in a easy way. The three main modes to use the service are:

- **Web interface**: This is the most common way to use it. It is a web user interface that is very intuitive. This is its main strong point. It allows the user to realize all the flow of steps in a very easy way.
- **Command Line Interface**: A command line tool call bigmler. It permits more flexibility than the web. It was never used, because we worked directly with the API.
- **API**: A RESTful API provided in many programing languages: Python, Java, Node.js, Clojure, Swift, Objective-C, C#, PHP.

The service can be used in development mode or production mode. The first one is free, but the drawback is the limitation of size tasks. The second one is a paid mode. Tthere are different plans, each one of them with its own characteristics.

2.3.1 Supervised Learning

BigML offers a huge variety of resources very useful for the user. In next subsections, its main supervised learning resources will be summarized.

2.3.1.1 Sources

Sources are the raw data for the problem under study. BigML accepts different formats file, but the most common used is a CSV. BigML also accepts as source, remotes files by a specific URL or files from specific servers. Once the source is upload, there is a range of possibilities to configure it: select the type of the features, the language of the source, the missing values management, or how has to be treated text or item features, are some of the available options (Figures 2.3 and 2.4).

[1] "Francisco J. Martín: data scientist es el trabajo más sexi y corto de la ...." 25 ene.. 2016, http://www.sorayapaniagua.com/2016/01/25/francisco-j-martin-data-scientist-es-el-trabajo-mas-sexi-y-corto-de-la-historia/.

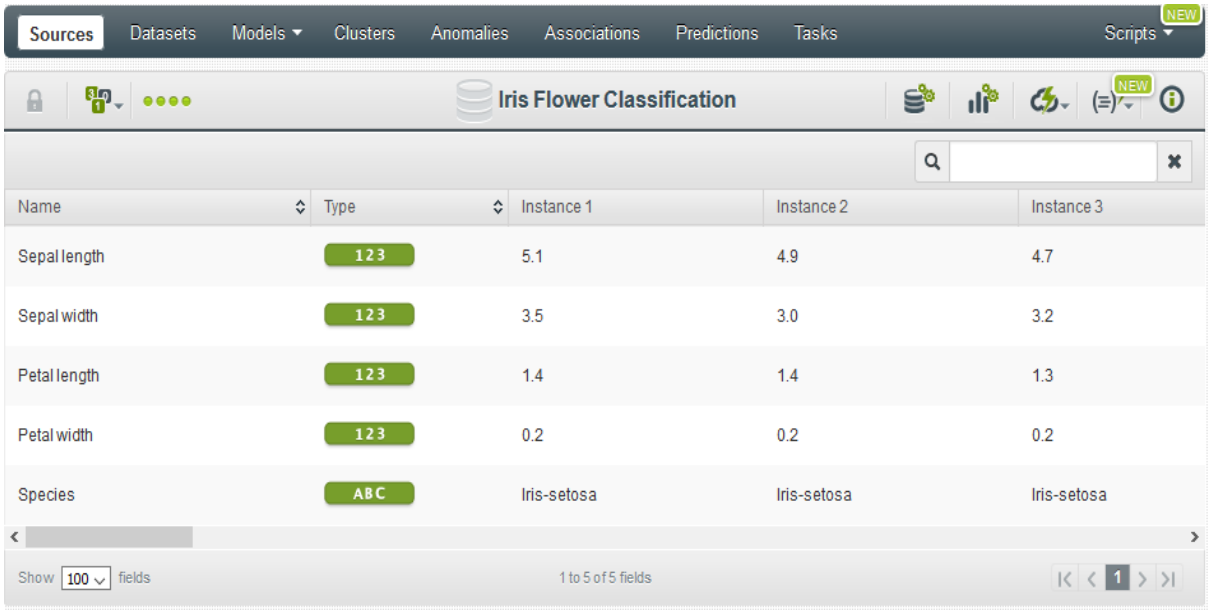


Figure 2.3: Source data.

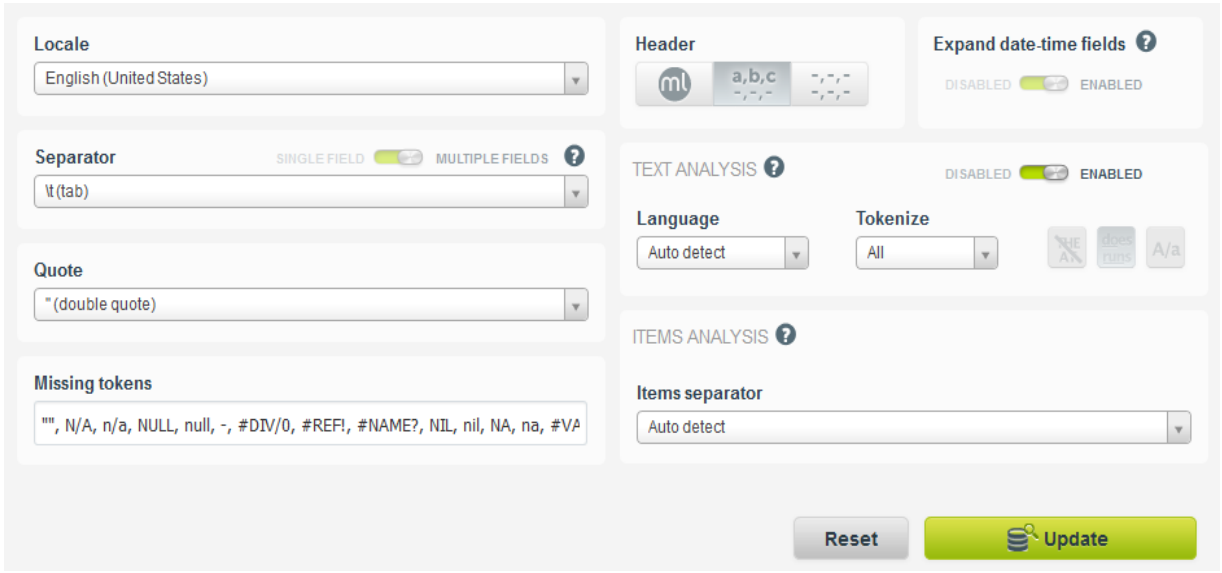


Figure 2.4: Source configuration.

2.3.1.2 Datasets

Datasets are views of the data source that the user can use as the basis for building models. Datasets specify the target attribute (class in classification or output in regression). Each feature is summarized with a bar graph that permits its visualization (Figure 2.5). In addition, the user can see some statistical descriptive values of the variables like the mean, median, standard deviation, etc. that permits a first analysis of the features' distribution.

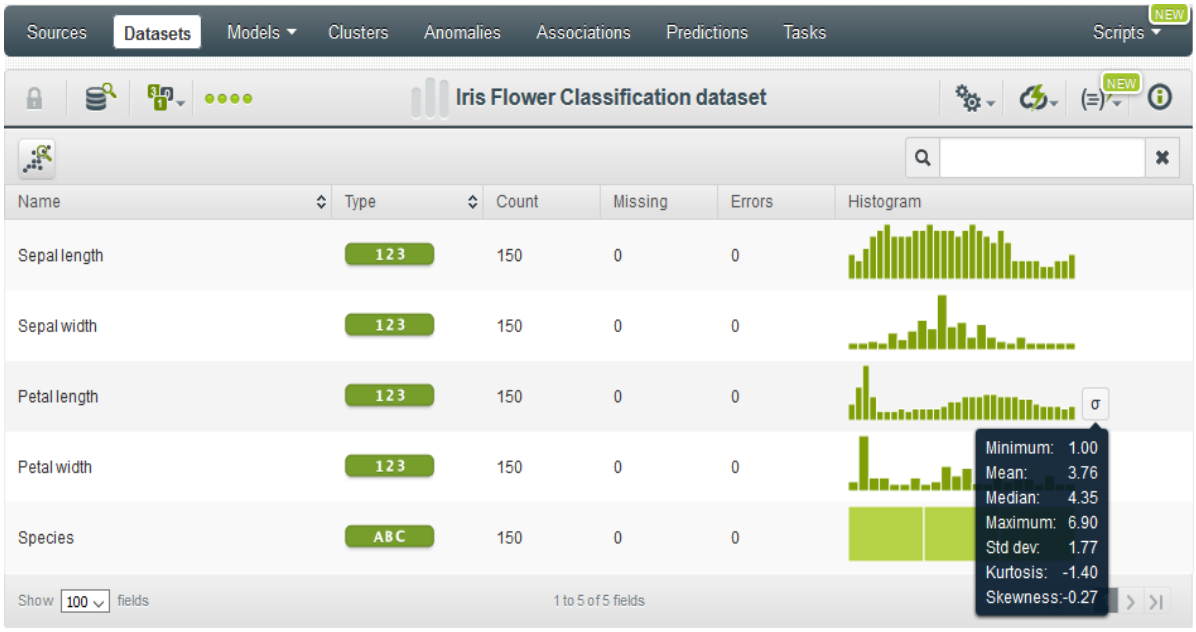


Figure 2.5: Features distribution.

There is also a very common used training and test set split resource that separate an original dataset into a training and test dataset for a controlled evaluation of models performance. The user can choose the proportion data of each set (see Figure 2.6).

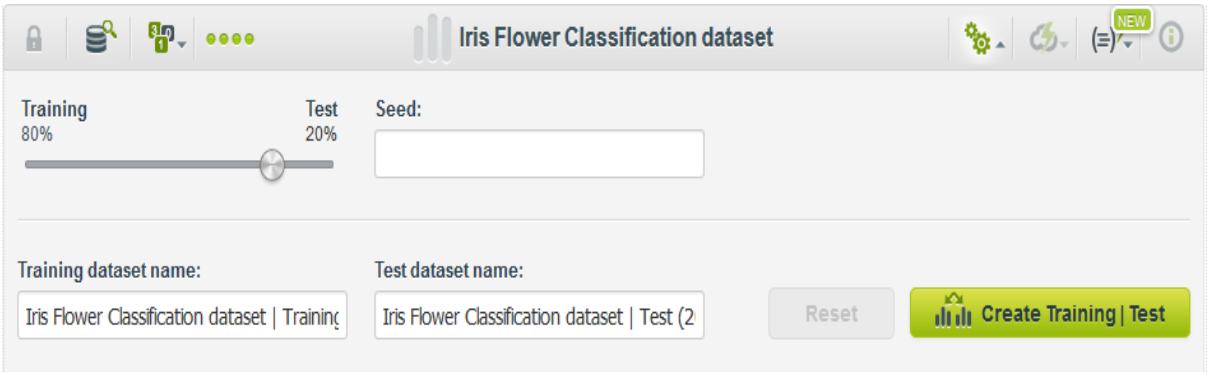


Figure 2.6: Split dataset.

2.3.1.3 Discriminant Models

A discriminant model, like a decision tree can be induced from a dataset. One of the best characteristics of BigML is the interactive interface it provides. The user can see the confidence and support in the training data reflected in the model at each node, and how the rules are build up, which is a clever and clean presentation of the model (Figure 2.7). BigML offers a sunburst view representation as well (Figure 2.8).

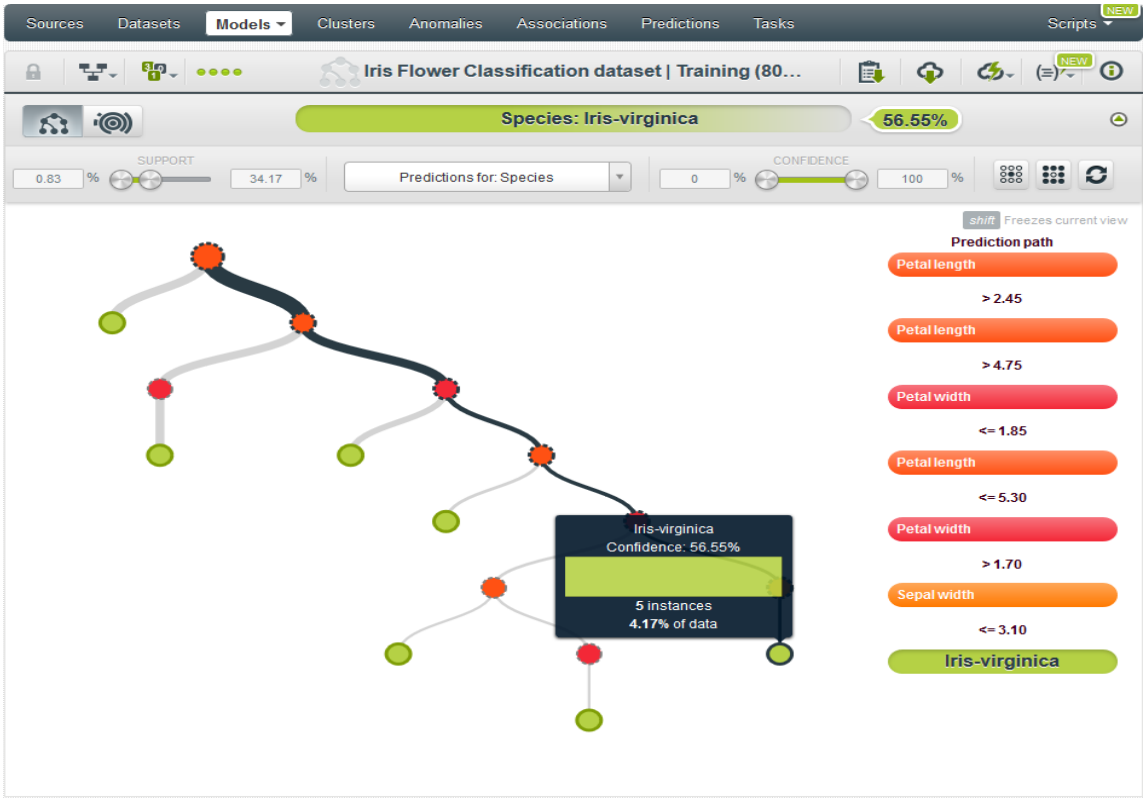


Figure 2.7: A Decision Tree Model.

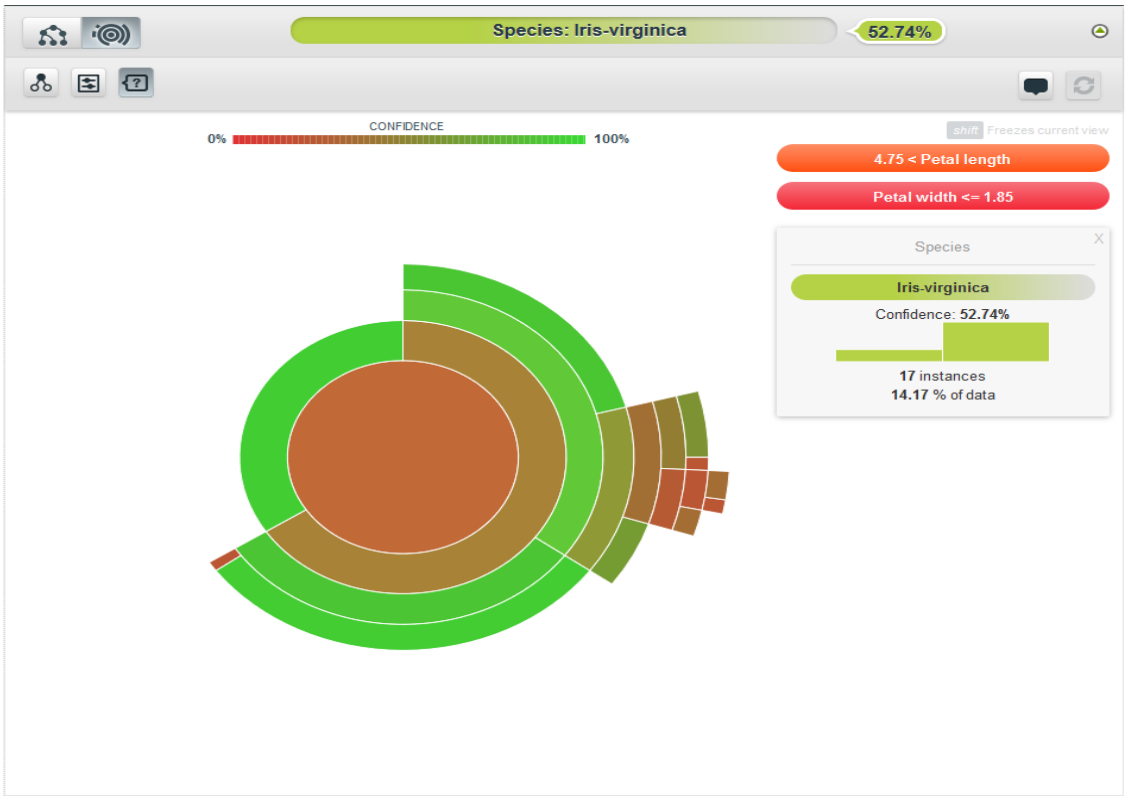


Figure 2.8: Sunburst view.

As in the different resources, models have its own properties that makes them flexible to the user demands. Some options as the balanced objective or the number of leafs are examples of the variety of parameters' configuration.

2.3.1.4 Ensembles

An ensemble is a collection of models which work together to create a stronger model with better predictive skills (Figure 2.9). BigML provide two type of ensembles configuration:

- **Bagging** (Bootstrap Aggregating): it builds each model from a random sampling the of dataset. By default, the samples are taken using a rate of 100% with replacement. This strategy often outperformsmore complex strategies.

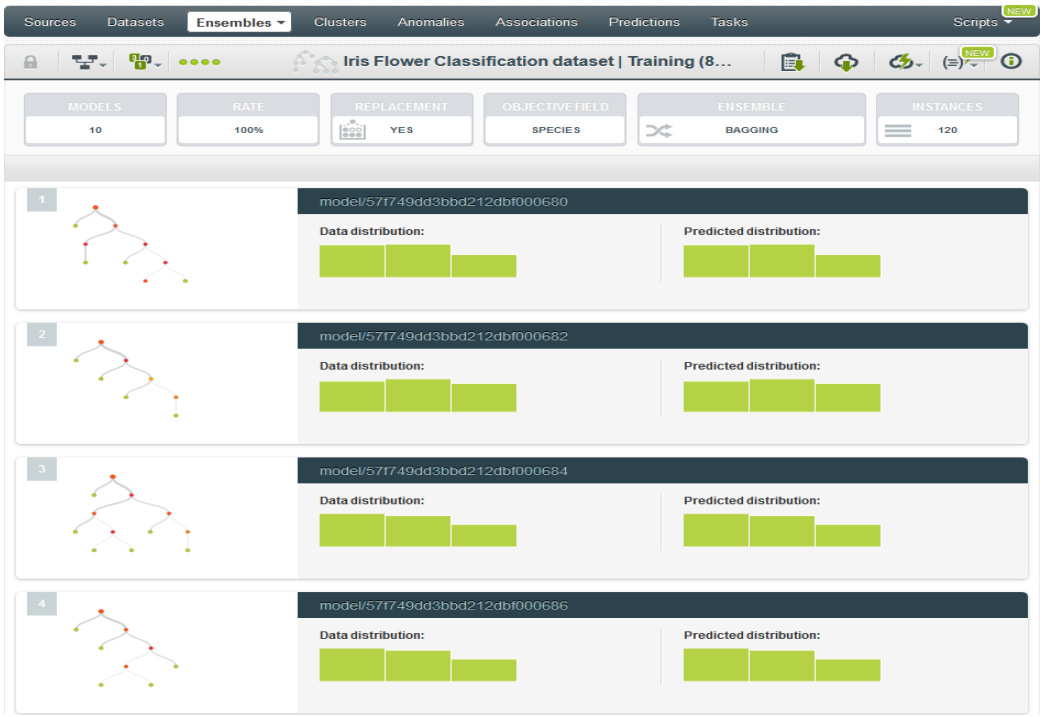


Figure 2.9: Ensemble.

- **Random Decision Forest**: a similar strategy to Bagging, with random sampling, but in addition, to build each decision tree, it chooses from all the available features, a random feature subset at each split.

Ensembles and models, once trained, have the option to visualize an ordered list with the field (attribute) importance (see Figure 2.10). This characteristic, is very useful for the users. As mentioned before, this visualization ability is very important to understand the results, and communicate the information to the users. Usually, the client wants brief and simple answers, like this kind of visualization.

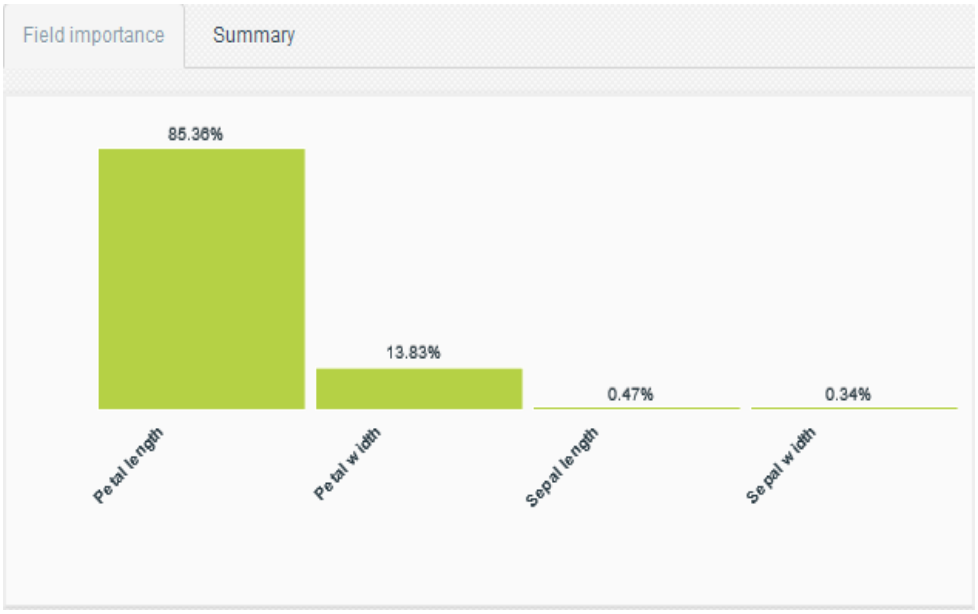


Figure 2.10: Field Importance.

2.3.1.5 Logistic Regressions

A logistic regression is a supervised Machine Learning method to solve classification problems. For each class of the objective attribute, the logistic regression computes a probability modelled as a logistic function value, whose argument is a linear combination of the field values (see Figure 2.11).

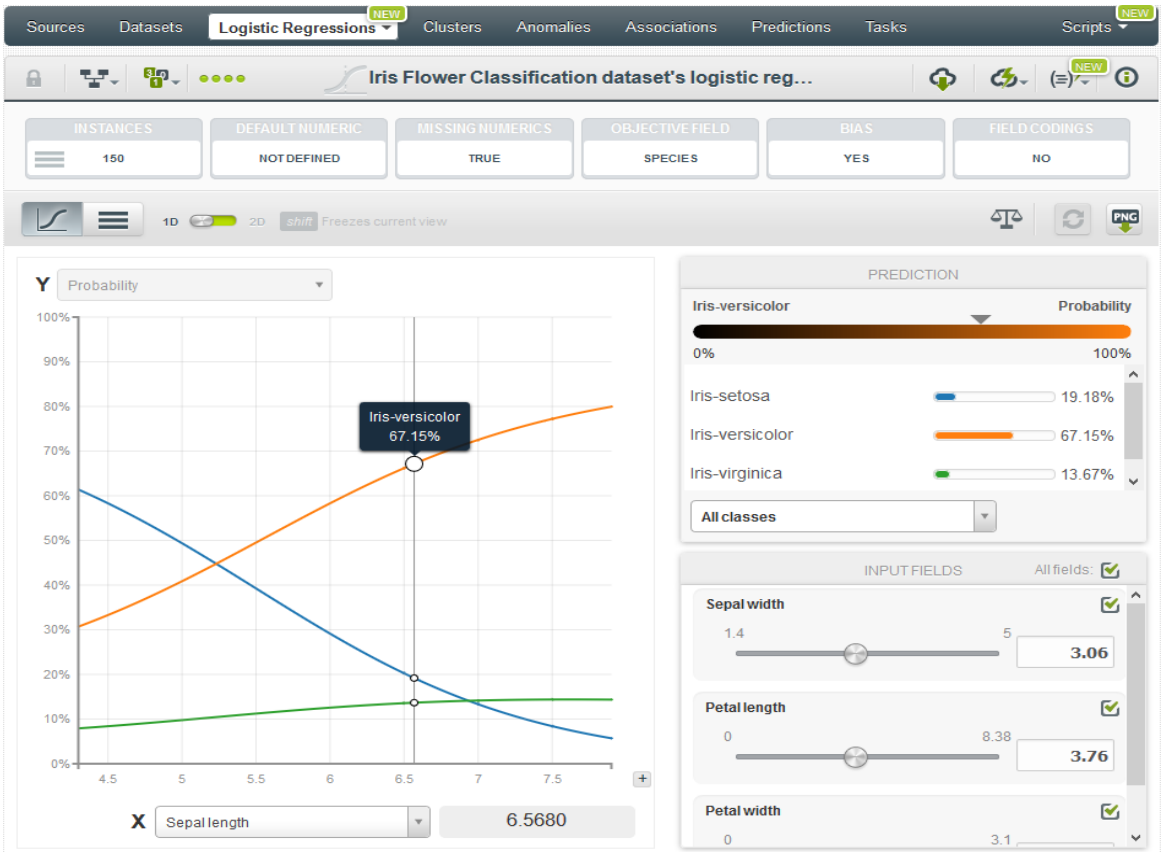


Figure 2.11: Logistic Regression.

2.3.1.6 Predictions

BigML permits predictions for single instances or for many instances in a batch (Figure 2.12). Each prediction has a categorical or numerical output depending if it is a classification/discriminant or regression/predictive problem respectively. In addition, for each prediction there is its confidence or expected error, respectively.

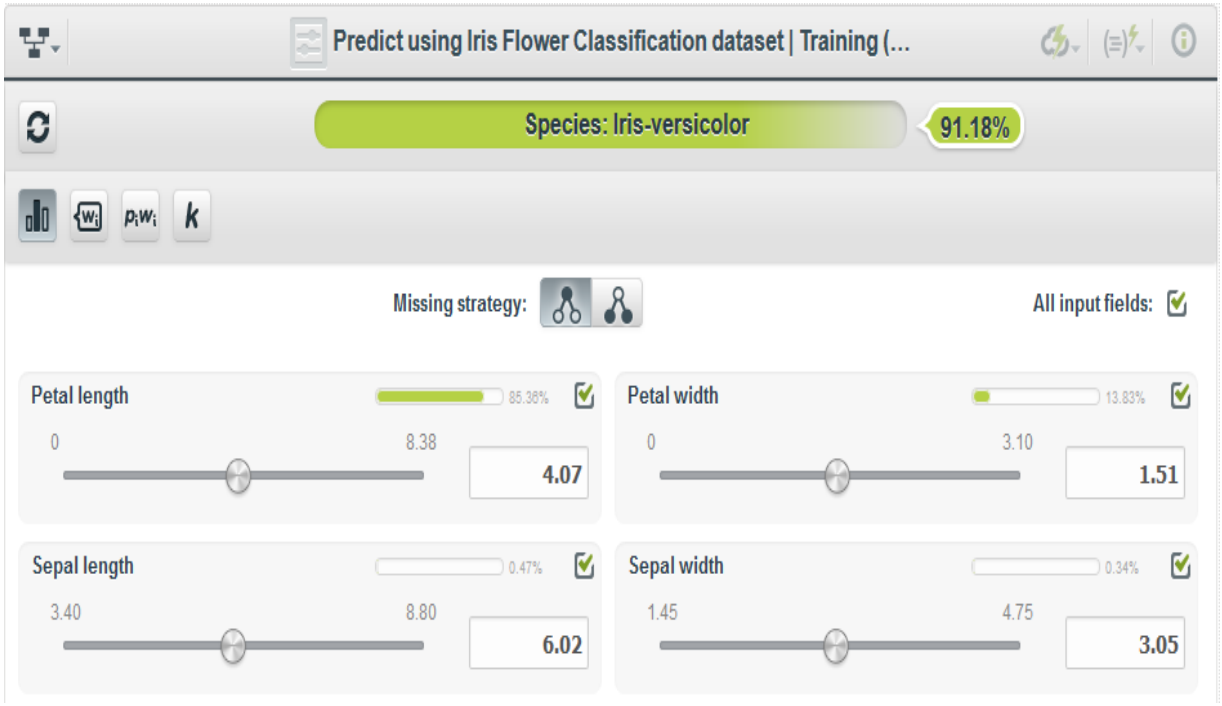


Figure 2.12: Single Prediction.

2.3.1.7 Evaluations

BigML provide an easy way to measure and compare the performance of classification and regression models. The main purpose of evaluations are:

- First, obtaining an estimation of the model's performance in production (i.e., making predictions for new instances the model has never seen before).

- Second, providing a framework to compare models built using different configurations or different algorithms to help identify the models with best predictive performance.

The basic idea behind evaluations is to take some test data, different from the one used to train the model and create a prediction for every instance. Then, the actual objective field values of the instances in the test data are compared against the predictions, and several performance measures based on the correct results as well as the errors made by the model are computed.

2.3.2 Unsupervised Learning

BigML offers a variety of unsupervised learning resources as well. As in the project, major models developed were unsupervised learning models; they will be more detailed than the supervised learning resources. However, these resources will be deeply explained in the next section.

2.3.2.1 Clusters

BigML Clusters provide powerful visualizations of the results of clustering data instances, which gives an insight into their internal structure. In addition, their visual representations of the clusters also provide a textual summary view of the most essential information about them (see Figure 2.13). Clusters uses proprietary unsupervised learning algorithms to group together the instances that are closer together according to a distance measure, computed using the values of the fields as input.

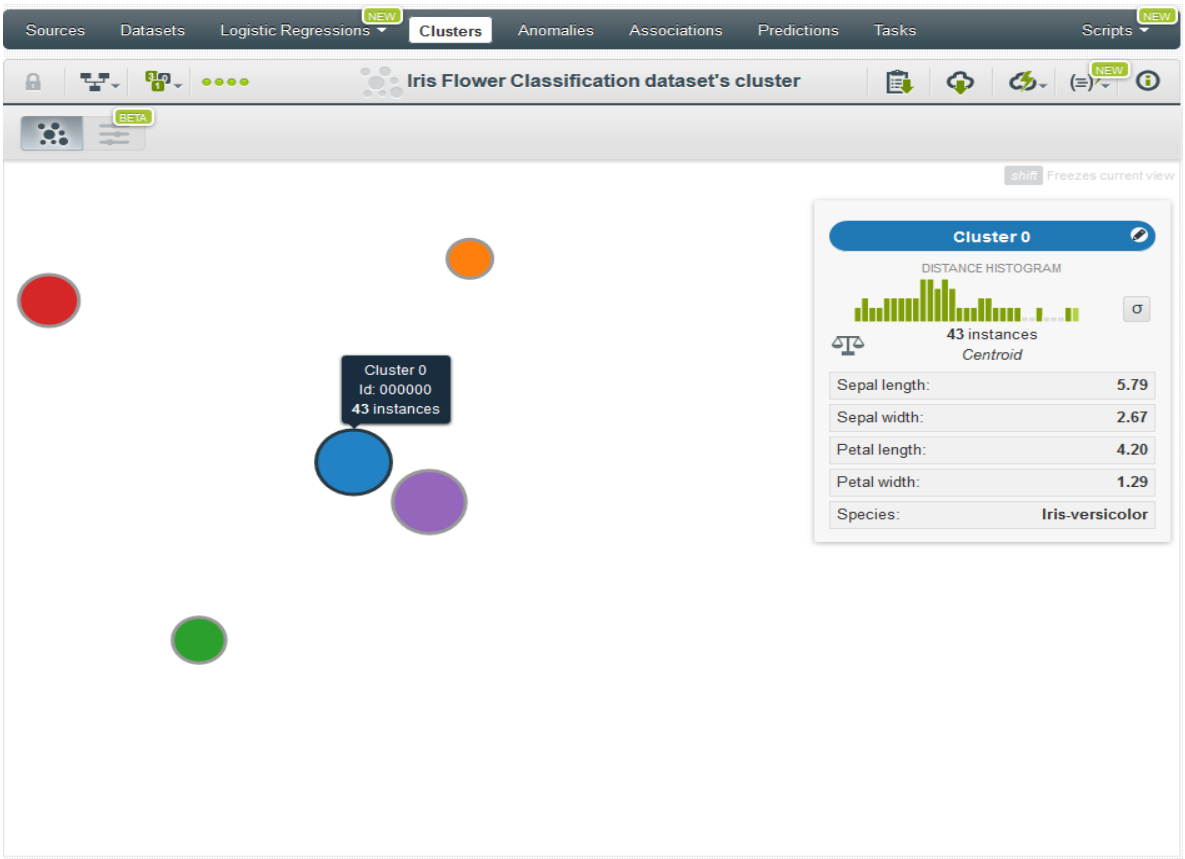


Figure 2.13: Clusters.

BigML Clusters can be built using two different unsupervised learning algorithms:

- **K-means**: the number of centroids need to be specified in advance.

- **G-means**: learns the number of different clusters by iteratively taking existing cluster groups and testing whether the cluster's neighborhood appears Gaussian in its distribution.

Both algorithms support a number of configuration options, such as scales and weights, over others.

2.3.2.2 Anomalies

This functionality allows identifying instances within a dataset that do not conform to a regular pattern (see Figure 2.14). BigML's anomaly detector is an optimized implementation of the Isolation Forest algorithm, a highly scalable method that can efficiently deal with high-dimensional datasets.

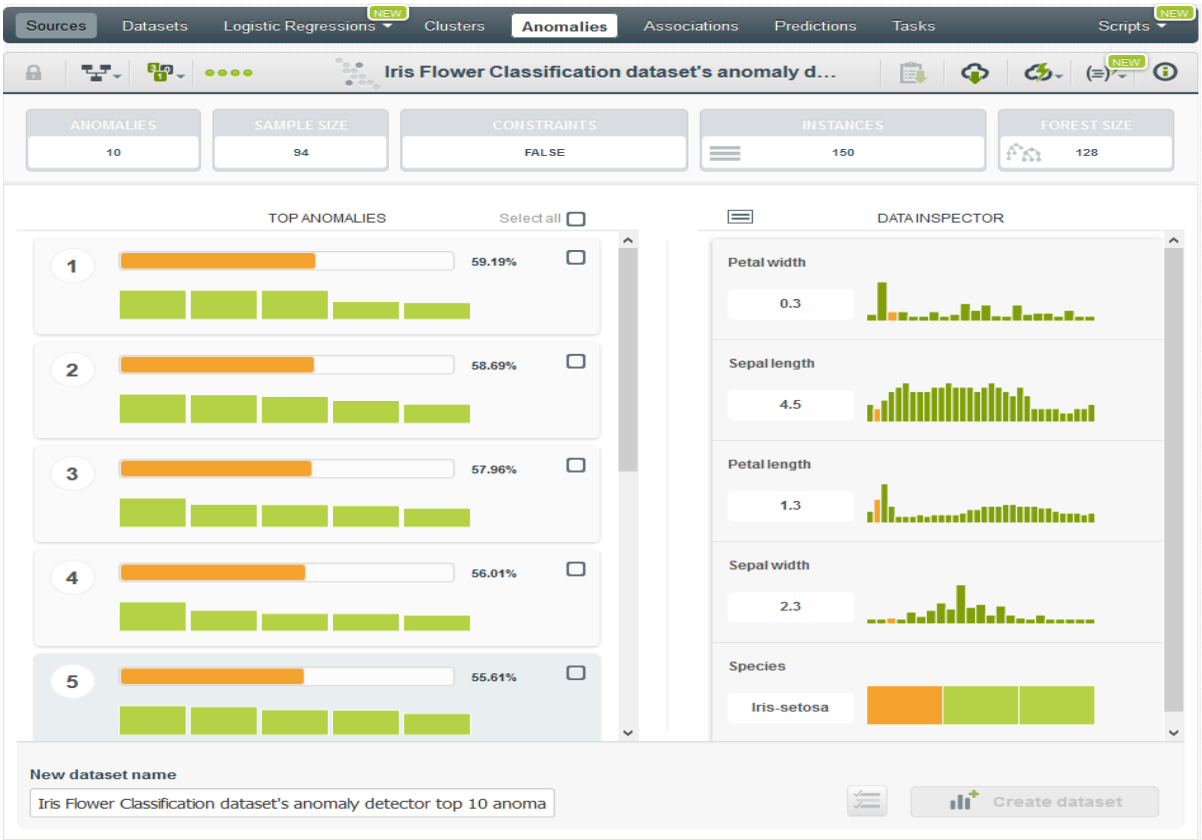
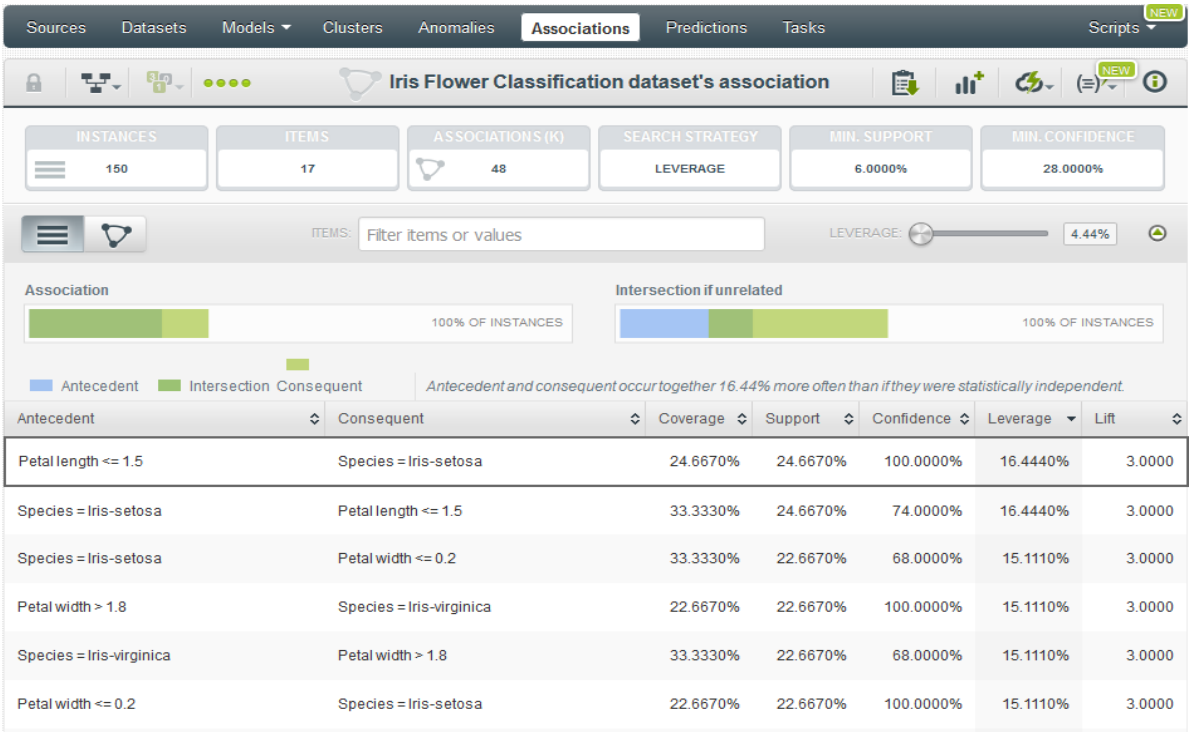


Figure 2.14: Anomaly Detection.

2.3.2.3 Association Rules

A functionality is available to discover meaningful relationships among fields and their values in high-dimensional datasets, using an association rules technique (see Figure 2.15).



Antecedent	Consequent	Coverage	Support	Confidence	Leverage	Lift
Petal length <= 1.5	Species = Iris-setosa	24.6670%	24.6670%	100.0000%	16.4440%	3.0000
Species = Iris-setosa	Petal length <= 1.5	33.3330%	24.6670%	74.0000%	16.4440%	3.0000
Species = Iris-setosa	Petal width <= 0.2	33.3330%	22.6670%	68.0000%	15.1110%	3.0000
Petal width > 1.8	Species = Iris-virginica	22.6670%	22.6670%	100.0000%	15.1110%	3.0000
Species = Iris-virginica	Petal width > 1.8	33.3330%	22.6670%	68.0000%	15.1110%	3.0000
Petal width <= 0.2	Species = Iris-setosa	22.6670%	22.6670%	100.0000%	15.1110%	3.0000

Figure 2.15: Associations rules.

Chapter 3

Design and Application of a Market Basket Analysis Methodology

3.1 Project Methodology

The aim of this project is the analysis of customers' purchases and its behaviour. To do it, the project was divided in three steps as it was described in the "*Introduction"* chapter. In this section, the methodology used will be described to provide a more specific idea how the different parts are performed.

The first step in this project was the creation of the dataset that summarized the stores' behaviour for the clustering process. Each row in the dataset was a store and columns were its features. Features were divided in three groups, structural, geographical, and behavioural. Structural features were information like the size of the store or whether it posses a parking. Geographical features were information like the city or the region. Behavioural features were information like the units sold per quarter or the income of the rent. All this information was obtained from the data the client provided. The construction of features was an iterative process where at the end of each iteration a set of features was obtained.

One point to remark was that stores that opened in a date, a posteriori of the tickets record start date and stores closed in a date previous at the tickets record end date were removed. On the first case, stores were removed to avoid training the clustering with stores that do not have a complete information over the whole historical year. These stores could add noise to the model performed. On the second case, stores were erased simply because they were already close. It has no sense to identify to which cluster belongs a store, if it does not already exists.

With the dataset created, an analysis of clustering algorithms was performed. Three clustering algorithms were used in this process, Hierarchical agglomerative, K-means and G-means. Hierarchical agglomerative algorithm used *Ward's* linkage criteria. With the G-means algorithm, in order to obtain the desired numbers of cluster this was achieved tuning the *critical value* parameter. The analysis process consisted in two parts:

A. **Clustering composition comparison**: The aim of this analysis was the evaluate whether different clustering algorithms, using the same dataset, could lead to similar clusters results. To do that, for each pair of clustering results, the two partitions generated (P_1, P_2), were analysed the stores that composed the clusters from the first partition coming from the first algorithm, and was compared to the stores that composed the most similar cluster from the other algorithm.

   To calculate the similarity between two clusters (C_1, C_2) it was used the *Jaccard coefficient*, which computes the intersection set of stores of both clusters divided by the union set of stores of both clusters. The value range is [0, 1], where 0 means no similarity between clusters and 1 both clusters are composed with the same set of stores. Higher values mean higher similarity between the two clusters.

$$ClusSim(C_i, C_j) = \frac{|C_i \cap C_j|}{|C_i \cup C_j|}$$

   To compute the similarity of the output of two different clustering algorithms (i.e., between two different partitions of the set of stores) was defined the *Aggregated Cluster Similarity (AggClusSim)* measure which computes the average cluster similarity between each pair of clusters which maximizes the cluster similarity. The value range is [0,1] where 0 means no similarity between the partitions and 1 means that both partitions are exactly the same, i.e, they have the same set of clusters clusters. Higher values means higher similarity between the two partitions.

$$AggClusSim(P_1, P_2) = \frac{1}{N}\sum_{i=1}^{N} \max_{C_j \in P_2 - \{C_j \in P_2 | \max_{\substack{C_k \in P_1 \\ 1 \le k < i}} ClusSim(C_k, C_j)\}} \max_{C_i \in P_1} ClusSim(C_i, C_j)$$

B. **Structural clustering analysis**: The aim of this analysis was the evaluation of which algorithms performed a "better" clustering. Unsupervised methods cannot be evaluated as supervised methods, because there is not a reference partition to compare with. Due that, in order to evaluate the clusters, cluster validation indexes were used. These indexes evaluate clusters based on its compactness, separation and the relationships between compactness and separation. The set of indexes used were, Minimum Cluster Separation (δ), Maximum Cluster Diameter (Δ), Dunn Index (D) and Davies-Bouldin Index (DB). For each algorithm, those indexes were evaluated using different number of clusters.

In addition, to evaluate the set of clustering results, the client's knowledge in the domain was used to analyse the clusters.

Based on the previous analyses, it was selected the best algorithm. With that, there were created the definitive clusters and the descriptive analysis of them. The aim of this analysis was to describe and compare the set of clusters based on a set of features. To do that, a Random Forest with 100 trees was built. Using the dataset of stores, a new column was added where its value was the name of the cluster where the store belonged. This new column was the label to predict. Once trained the ensemble, the most important attributes were selected.

The last step was the creation of the association rules for each cluster. The methodology used in this process consisted in finding the association rules for the nearest store to the cluster centroid (i.e., the medioid). Then, those rules were extrapolated to all the other stores that composed that cluster. The measures used to generate the rules were Lift and Leverage.

To conclude, a web page was created were the results were published.

3.2 Software & Hardware used

The software used in this project was the programming language Python [Python, 2017] and BigML. In addition, it was used one of the most powerful libraries used nowadays for Data Science, Pandas [Pandas, 2017] and scikit-learn [Scikit-learn, 2017]. The programing environment used was Jupyter Notebook [Jupyter-Notebook, 2017].

For the feature engineering task, a Windows server was used. There, all the data was processed. For all the analysing tasks like the algorithm comparison, metrics and association rules, it was used a laptop.

Server

- Windows Edition: Windows Server 2012 R2 Standard
- Processor: Intel(R) Xeon(R) CPU E5-2047 v2 @ 2.40GHz 2.40 GHz
- Installed memory (RAM): 24.00 GB
- System type: 64-bit Operating System, x64 based processor

Laptop

- Windows Edition: Windows 10 Enterprise
- Processor: Intel(R) Core(TM) i7-5500U CPU @ 2.40GHz 2.40GHz
- Installed memory (RAM): 8.00GB
- System type: 64-bit Operating System, x64 based processor

3.3 Data Description

The key aspect in any data science project is the data. Data is the principal component that makes a project successful or failed. It is the machine learning algorithm combustible. Companies usually have its data in a data warehouse [Data warehouse, 2017] or databases [Database, 2017] and the extraction of it is a difficult task that requires a huge work.

In this project, the data used was composed by three different datasets: Tickets record, Items summary and Stores summary. Each of them was a *csv* file containing specific information about the business.

First, the Tickets record was a dataset formed by all the tickets casted through a year. On the second place, Items summary was a dataset formed by all the items the client had in stock and its characteristics. To conclude, Stores summary was a dataset formed by all the stores of the client and its characteristics.

The volume of data in this project was a constraint through all the process. In order to obtain the desired datasets, one for the clustering task and another for the analysis of association rules, it was needed a complex ETL (Extract, Transform and Load) or feature engineering [ETL, 2017]. Through the feature engineering process, it was created, transformed and cleaned features that could capture and represent the behaviour of the stores. The next figure 3.1 is a scheme of how data was integrated in order to create the dataset used for the clustering process.

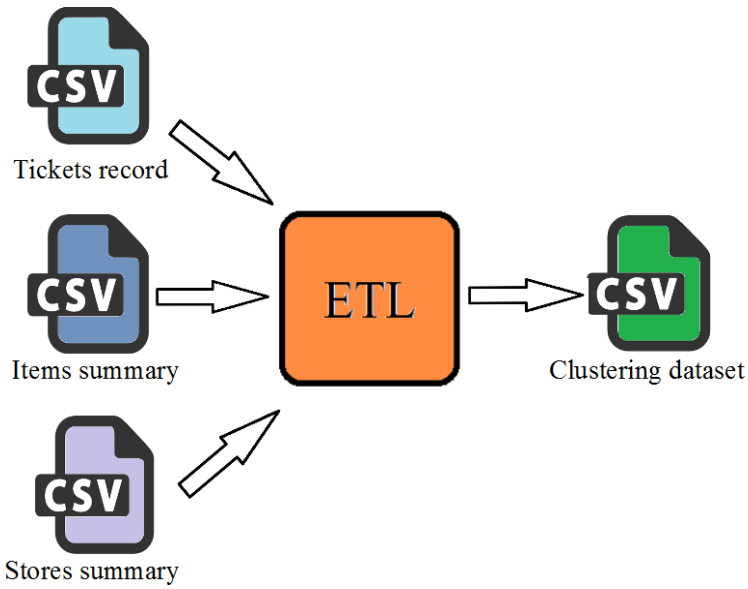


Figure 3.1: ETL or Feature Engineering scheme

In the following sections, each dataset will be described. For each of them, it will be a description. In addition, there will be a table summarizing the features the dataset had.

3.3.1 Tickets dataset

- Number of Instances: 214,712,174
- Number of Attributes: 13
- Missing Values? Yes
- Size of Dataset: 21.4 GB

The first dataset used in this project was the historical tickets record. It was composed by 36,763,526 tickets from 203 different stores. Those tickets were expended through June 2015 until May 2016.

A setback faced with the dataset was the size of it. Habitually, upload data to the resource where has to be processed is tedious task in some projects. There are tools like Hadoop [Hadoop, 2017] and Spark [Spark, 2017] that process huge volumes of data. However, in this project it wasn't needed the use of these tools. In order to provide the tickets records, the client split the historical in twelve pieces, one for each month. With that, the client reduced its size and could send the set of pieces via internet. Once collected all the set of different months, they were concatenated. The next figure 3.2, is a scheme of the process.

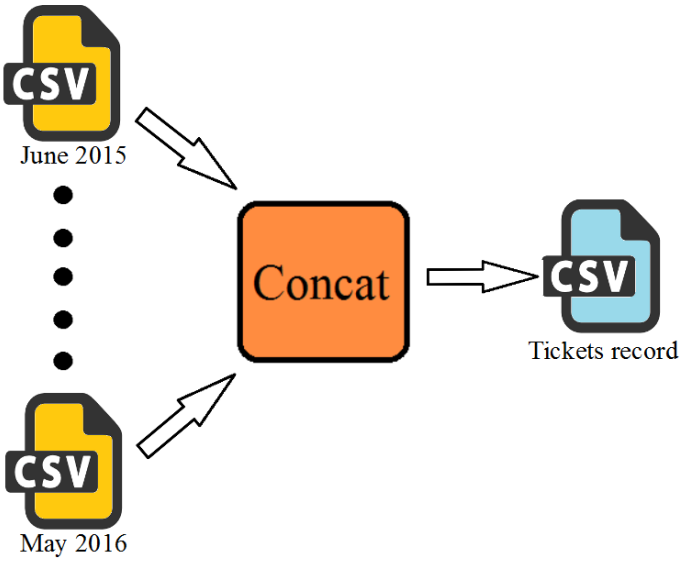


Figure 3.2: Tickets record dataset creation.

The structure of the dataset was the following: each row corresponded to an item purchased, and columns were features like the day it was purchased, the store where it has bought or the ticket it belonged to. For instance, using as guide the next image (Figure 3.3), it can be seen that the ticket with code "*2015060100000170200014*" is composed by 4 different products, the "*201054*", "*833950*", "*950025*" and "*950095*". This example corresponds to the lines [3-6] from the figure. The total set of features of the dataset is listed in the figure A.1 from the annex A.

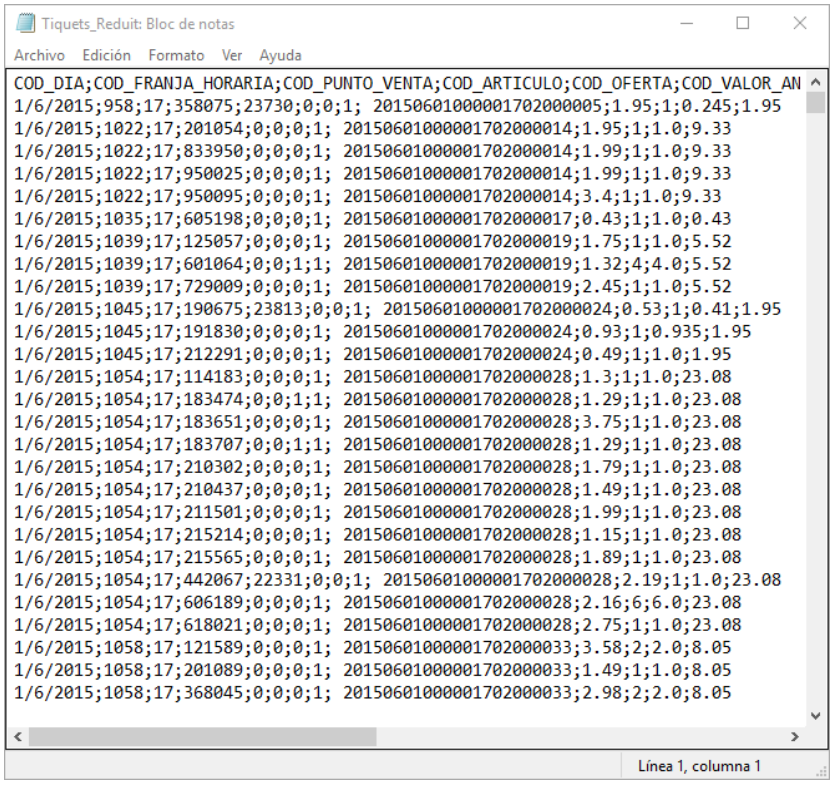


Figure 3.3: Tickets dataset screenshot.

3.3.2 Items dataset

- Number of Instances: 60,587
- Number of Attributes: 74
- Missing Values? Yes
- Size of Dataset: 65.5 MB

The second dataset used in the project was the stock of items. This dataset contained information of each item like the family, if it was ecologic or if it had gluten. Each instance was an item and the columns were its features.

To avoid overextending the summary table, some features were merged into one. For instance, most of the features in the database were repeated twice, one had the code and the othe,r the description or features were repeated in Catalan and Spanish. Due this overlap of information, in the figure A.2 from the Annex A are just listed a set of features that represented the concept.

3.3.3 Stores dataset

- Number of Instances: 273
- Number of Attributes: 73
- Missing Values? Yes
- Size of Dataset: 226 KB

The last dataset used contained information of each store. This information belonged to structural and un-structural variables like the size of the store, the location of it or the shop category it belonged to. The figure A.3 from the Annex A is the list of variables.

The hypothesis was that shops could be similar due to structural. For instance, a shop with parking could have higher mean ticket because people go there on car and can take with him more products. Moreover, the category of the shop implied to have some specific products that others do not have. All these features could influence in the behaviour of the shop.

3.4 Application of the Methodology

3.4.1 Data Pre-processing

Data pre-processing is the first step that must be performed in any data mining project. This process is vital for the project and it consists of finding inaccurate data like out-of-range values (e.g. units sell: -5), impossible data combinations (e.g. Stores' region: Madrid, City: Barcelona), missing values, incorrect columns name, empty field, etc. If there is much irrelevant and redundant information present, or noisy and unreliable data, then knowledge discovery phase is more difficult. This step take a considerable amount of time.

In this project, an exhaustive analysis was performed to understand the composition of the data and evaluate the quality of it. As it was described previously, data was divided in three blocks: Tickets dataset, Items dataset and Stores dataset. Each of these datasets was analysed at detail due its information was the one used to create the clustering dataset.

To perform data mining projects is it not strictly necessary a huge amount of data. It is quality, rather than quantity, what counts. Without quality, the algorithm cannot learn good patterns from data because the data could not be representative. Data usually have inconsistencies; some variables are not well calculated or others are obsolete. All these errors influence negatively the learned model. In this project, data had incorrect information as column headers with *Null* as name, or products identifiers with *Null* (both errors were found in the Items dataset). Those features or products were removed because the client for testing purposes used this information.

There are cases where data is correct but should be removed because it does not follow a common pattern. This data is considered an anomaly. Anomaly detection [Anomaly

detection, 2017] is the identification of observations, which do not conform to an expected pattern or other items in a dataset. This concept has not to be confused with data cleaning, because the cleaning data process search for invalid records. Anomalies detection search for instances that not form part of a pattern. Depending on the objective of the project, these anomalies can be noise or be exactly what you are looking for. For instance, in fraud detection problems, those instances that do not conform a regular pattern are possible fraudulent transactions [Dal Pozzolo & Bontempi, 2015].

In this project, the store with id *614* had 2500m^2, while the second store with more meters was the *525* with 1400m^2. The store *614* was so big due the client considers as a single store an entire commercial centre. This store profile does not follow a reasonable pattern, and thus, it can be considered as an anomaly store. The issue was to decide whether to remove or not the store from the clustering process. At the end, the store was not removed because the client was interested in knowing in which cluster, the clustering algorithm classified it.

In addition, to perform valuable results, through the data cleaning process of the tickets record dataset, plastics bags and parking records were removed because they did not add valuable information to the project.

3.4.2 Feature engineering

Once analysed the data, the next step of the project was the creation, transformation, and cleaning of the features. The aim of this process was the generation of a final dataset that captured the behaviour of the stores for the clustering process. Each row in the dataset was a store and columns were its features. To obtain the final dataset, this process was developed in successive versions, where on each of them, the result was a set of features.

Through this section, some of the most remarkable versions obtained through the feature engineering process will be described. As during the clustering process, a large proportion of versions were created applying just little changes into the features, there will not described all the different versions, just the ones most remarkable or interesting. For instance, when the type of a feature was changed or a correction of a mistake was done. This cumulative work affects the result, because this sum of work, at the end, is the one that add value to the project. For each version, it was obtained a dataset where the instances were the stores and the features were the store features.

All the versions were constructed to capture the temporality of the season. The hypothesis was that customer behaviour changed through the year. To capture this change, datasets were created with features repeated for each quarter. As the tickets record started from June 2015 until May 2016, the quarterly features were created according to these groups: {June 2015, July 2015, August 2015}, {September 2015, October 2015, November 2015}, {December 2016, January 2016, February 2016} and {March 2016, April 2016, May 2016}. This set of features contain a *"Trimestre X"* at the end of the name, where X is a number ranging from 1

to 4 according to these respective groups. Moreover, those stores that were opened or closed throughout the period were removed.

Through the analysis of each version, there will be a description of each feature used. Some features are simply obtained from the data the client provided; others are created during the feature engineering process. In addition, those features that need a deeper description to understand its function will be described in detail. As some features can be used in various versions, the description of them will not be repeated, just the new features used in the version will be described.

3.4.2.1 Features version 1

On the first version, most of the features were created. Some of them were obtained directly from the original data and others were created. The most interesting feature created in this version were the ones that described *the distribution of sold items for each section*. This concept in market analysis is known as *market share*. These features are the ones in the range *[27-53]*.

Market share [Market Share, 1999] is the percentage of an industry or market's total sales that is earned by a company over a specified period. Market share is calculated by taking the company's sales over the period and dividing it by the total sales of the industry over the same period. This metric is used to give a general idea of the size of a company in relation to its market and its competitors.

The figure B.1 from the Annex B is the list of features obtained in this version.

Description of new features added in this version:
Features [1]: Id of the shop.
Features [2]: Price of the most expensive ticket.
Features [3]: Mean price of all tickets.
Features [4-6]: Mean units sold per day, week and month.
Features [7-13]: Which percentage of items from total are sold on each weekday. The sum of all this value is up to 100.
Features [14]: Number of items sold from the client's brand.
Features [15]: Number of items sold from other brands.
Features [16]: Number of different items the shop has.
Features [17]: Relation between the number of items sold from the client's brand vs the number of items sold from others brand.
Features [18]: Relation between the number of items sold.
Features [19]: Mean number of items sold per ticket. Mean number of references per ticket.
Features [20]: Mean number of references per ticket.
Features [21-23]: The first item appears most times in tickets, the second one, and the third one.

Features [21-26]: The first item most sold, the second one, and the third one.
Features [27-53]: Which percentage of items from total are sold from each section. The sum of all this value is up to 100.
Features [54-58]: The shops possess the corresponding counter.
Feature [59]: Segmentation created by the client to classify the shops.
Feature [60]: Segmentation created by the client to classify the shops by their size.
Feature [61]: The shop close at midday.
Features [62-65]: Shop's Geolocation
Feature [66]: Shop's surface in m^2

3.4.2.2 Features version 2

In this version were changed the features used for the market share. In addition, new features were added to have more information about *stores' revenue*. The hypothesis was that revenue could be a good indicator about stores behaviour. For that reason, features capturing this information were created.

To capture the stores' revenue a new feature was created: *Items 80/20*. This new information was the list of items that added most revenue to the store. This feature was created based on the Pareto principle [Pareto principle, 2017]. In retail domain, the Pareto principle represents that just a short list of different items adds most of the revenue. For instance, the baguette is an often purchased item. In addition, it is an item that people often buy more than one unit at a time because is highly used. On the contrary, the broom is an item that is sold once and until it is not useful is not bought again. To do that, items purchased were ordered by its revenue until reach the sum of 80% of total revenue.

The figure B.2 from the Annex B is the list of features obtained in this version.

Description of new features added in this version:
Features [27-42]: In the previous version, the features used for market share were done at section level. Now this feature is constructed at family level.
Features [52]: This is the revenue of the shop for each quarter.
Features [53]: Relation between the revenue and size of the shop. As higher is the number, better is.
Features [54]: List of 20% items that add 80% of the total revenue. This feature was created based on the Pareto principle. This principle says 80 percent of the outcomes come from 20 percent of the inputs.

3.4.2.3 Features version 3

In this version were added two new concepts in the dataset; *the revenue of each section* and *the market penetration rate*. Both concepts were important to obtain the final dataset and represented an important point of view from the client side.

Market penetration rate [Market Penetration, 1999] and market share rate go hand in hand as metrics descriptions in retail. These features represent which presence has each section on the shop's tickets. For instance, if a shop has 4 different tickets, and on 3 of them have, at least, one item of a specific section, the penetration rate of that section will be 75%.

The figure B.3 from the Annex B is the list of features obtained in this version.

Description of new features added in this version:
Features [27-42]: Revenue for each section
Features [59-74]: Penetration rate for each section.

3.4.2.4 Features version 4

In this version were not added new features. The changes done in the dataset were the elimination of some features and the rename of others to make them easier to read. The list of features erased were *Tipo tienda client, Talla centro, Municipio, Codi postal, Max Ticket (for each quarter), TOP Referencia aparece en mas tickets (for each quarter and number), TOP Referencia se venden mas unidades (for each quarter and number).* In addition, the feature *Items 80/20 - Trimestre X* is no longer created based on items is based on families instead.

The figure B.4 from the Annex B is the list of features obtained in this version.

3.4.2.5 Features version 5

This was the last version of features created. It was decided to be the last version because the way the current dataset was constructed, described enough well stores' behaviour. In addition, the deadline of the feature engineering task was near and it was impossible to invest more time creating new features.

The most important change in this version was the transformation of the features *Penetracio %*. These features were no longer created for each quarter. They were created for all the year. Moreover, they were not on section level. They were on family level instead. To conclude, the name changed to *Coverage %*. As there is a huge number of different families, on the list of features is just written *Coverage % Families*. The total number of features used was 395.

The figure B.5 from the Annex B is the list of features obtained in this version.

3.4.3 Clustering Techniques

After the feature engineering step, which was obtained a definitive dataset of stores, started the store clustering process. The goal was to obtain different groups or clusters of the different stores to discover and characterise different profiles of stores. The first problem was the selection of the clustering method. As detailed in section 2.2.1, in the literature there are several clustering algorithms, which can be used.

In next subsections, the clustering technique selected is detailed, and the *sensitivity analysis* undertaken through an extensive comparison among several clustering methods is presented. The aim of this analysis was to examine the variations in the results (composition of the clusters) which could be caused by the selection of one clustering method or another. Our hope was that the selected clustering technique should not affect the resulting clusters or groups. Furthermore, a structural validation of the obtained clusters was done to test several structural properties of the clusters like compactness, separation, etc.

3.4.3.1 Selection of the Clustering Technique

A priori, the G-means method was a candidate technique to be selected, because it was an appealing method, based on the common *k*-means method, where the number of clusters, *k*, is determined automatically. However, as previously explained, there are several methods that can be applied. Hence, it was thought that some kind of sensitivity analysis should be done in order to compare different clustering algorithms, and to check whether the results were somewhat similar or not (clustering composition comparison) and in addition, a structural validation analysis of the clusters should be done.

3.4.3.1.1 Clustering composition comparison

To analyse the sensitivity of the clustering process, a comparison among clustering algorithms was done. The aim of this analysis was to compare if using the same dataset with different clustering algorithms, the output clusters were rather similar or not. If the clusters results differed enough between them, which would mean that the dataset is sensitive to variations. However, if the cluster results were rather similar among the clustering algorithms that would mean that the dataset is robust to small variations.

To perform the analysis three different algorithms, coming from different kind of clustering techniques were selected:

- A hierarchical agglomerative clustering technique (using Ward's method)
- A partitional clustering technique with fixed number of clusters (K-means)
- A partitional technique with automatic determination of the number of clusters (G-means)

For the experimental comparison, the number of clusters used was set to 5 and 9. Thus, the algorithms were run twice for obtaining 5 or 9 clusters. The obtained clusters were compared by each pair of algorithms. The Cluster Similarity measure (*ClusSim*) and the Aggregated Cluster Similarity (*AggClusSim*) measure described in the previous section were used. The results of the comparison are depicted in the six tables (see Table 3.1 to Table 3.6). The composition of the tables is the following: the first two rows are the algorithms used together with the id of the clusters obtained. The next two rows are the intersection and union of the stores for each pair of clusters maximizing *ClusSim*. The last row is the ratio between the Intersection and the Union, i.e, the value of the Cluster Similarity measure *ClusSim* for each pair of clusters which maximizes the similarity in the right order comparison. Finally, the last column of the last row is the sum of the maximizing *ClusSim* values divided by the number of clusters, which is the value corresponding to the Aggregated Cluster Similarity (*AggClusSim*) measure, described in the previous section. It gives a global measure of the composition of all clusters for the two partitions resulting from the execution of the two clustering algorithms.

The next three tables depict the results of the algorithm comparison with 5 clusters (Tables 3.1, 3.2, 3.3).

Table 3.1: Comparison between K-means 5 and G-means 5 methods.

	Cluster id					
K-means 5	0	3	1	4	2	
G-means 5	0	3	4	2	1	
	Cluster Similarity					
Intersection	11	28	51	41	0	
Union	11	36	76	87	45	***AggClusSim***
Max ClusSim	1	0.78	0.67	0.47	0	**0.58**

Table 3.2: Comparison between K-means 5 and Hierarchical Agglomerative 5 methods.

	Cluster id					
K-means 5	3	2	1	4	0	
Hierar. Agglom. 5	3	4	2	1	0	
	Cluster Similarity					
Intersection	27	16	44	33	11	
Union	29	25	71	89	41	***AggClusSim***
Max ClusSim	0.93	0.64	0.62	0.37	0.27	**0.57**

Table 3.3: Comparison between G-means 5 and Hierarchical Agglomerative 5 methods.

	Cluster id					
G-means 5	3	2	4	1	0	
Hierar. Agglom. 5	3	1	2	0	4	
	Cluster Similarity					
Intersection	27	49	39	20	0	
Union	35	79	67	42	28	***AggClusSim***
Max ClusSim	0.77	0.62	0.58	0.47	0	**0.49**

With the number of clusters equal to 5, the clusters results are quite similar between algorithms. The most similar algorithms are the K-means and the G-means with a value of 0.58. K-means and the Hierarchical Agglomerative are similar with a value of 0.57 as well. The most dissimilar clusters are the ones of the G-means and the Hierarchical Agglomerative.

The next three tables depict the results of the algorithms comparison with 9 clusters (Tables 3.4, 3.5, 3.6).

Table 3.4: Comparison between K-means 9 and G-means 9 methods.

	Cluster id									
K-means 9	0	2	7	8	3	5	1	6	4	
G-means 9	8	2	0	4	5	6	1	3	7	
	Cluster Similarity									
Intersection	28	24	9	8	30	9	18	13	1	
Union	29	28	11	13	55	18	42	31	19	***AggClusSim***
Max ClusSim	0.97	0.86	0.81	0.61	0.55	0.50	0.43	0.42	0.05	**0.58**

Table 3.5: Comparison between K-means 9 and Hierarchical Agglomerative 9 methods.

	Cluster id									
K-means 9	0	5	7	4	1	2	6	3	8	
Hierar. Agglom. 9	3	4	2	5	8	6	7	0	1	
	Cluster Similarity									
Intersection	27	14	8	9	16	18	15	26	9	
Union	28	18	12	14	25	30	31	54	32	***AggClusSim***
Max ClusSim	0.96	0.78	0.66	0.64	0.64	0.60	0.48	0.48	0.28	**0.61**

Table 3.6: Comparison between G-means 9 and Hierarchical Agglomerative 9 methods.

	Cluster id									
G-means 9	8	3	2	0	1	6	5	4	7	
Hierar. Agglom. 9	4	7	2	3	5	6	1	0	8	
	Cluster Similarity									
Intersection	27	13	18	9	20	9	21	2	0	
Union	29	15	26	13	43	20	55	49	17	***AggClusSim***
Max ClusSim	0.93	0.86	0.69	0.69	0.46	0.45	0.38	0.04	0	**0.5**

With a number of clusters equal to 9, the clusters similarity results increased, in general. On K-means and G-means algorithms, the values did not change. However, in K-means and Hierarchical Agglomerative algorithm the value increased in 0.04 points. In addition, G-means and Hierarchical Agglomerative algorithm increased as well in 0.01 points.

For both number of clusters, the results obtained indicated that independently of the clustering algorithm used, the clusters composition could be considered as quite similar. In addition, as more clusters were obtained, most similar were the clusters composition.

Even though this kind of analysis is not common in the literature, from a brief survey done applying this measure to other datasets, the *AggClusSim* when comparing different partitions coming from different clustering methods usually gives low values, less than 0.4. Therefore, the values obtained in this study were rather satisfactory regarding the sensitivity of the composition of the clusters to a small variation. Independently of the clustering technique used, the composition of the clusters is quite similar.

3.4.3.1.2 Structural Validation through Cluster Validation Indexes

The validation of a partition (set of clusters) is commonly done in the literature using one or more Cluster Validation Indexes (CVIs). As described in 2.2.3.1 there many CVIs proposed in the literature, but they are measuring different properties. Following the guidelines outlined in [Sevilla-Villanueva *et al.*, 2016], the most important properties are the *compactness of the clusters*, the *separation of the clusters* and the *relationship between compactness and separation*. In order to analyse the structural quality of the clusters, four cluster validity indexes were selected:

- Minimum Cluster Separation (δ), which measures the separation
- Maximum Cluster Diameter (Δ), which measures the compactness
- Dunn Index (D) and Davies-Bouldin Index (DB), which measures the relationship between compactness and separation

The aim of this analysis was the detection of which algorithm, using different number of clusters, obtained the "best" partition (set of clusters) from a structural point of view according to the selected metrics. It is needed to outline that the best structural partition does not guarantee to have a useful and interpretable partition. The proper way to complement a structural validation analysis is to make a validation and interpretation process by the domain experts. The metrics are just a tool to decide whether the set of clusters are structurally "well" defined.

The analysis was performed with the same set of algorithms than in the previous analysis (K-means, G-means and Hierarchical agglomerative clustering with Ward's method) and varying the number of clusters according to these values: {2,5,9,14}. For Minimum Cluster Separation (δ), high values are better, which means a higher separation among the clusters. For Maximum Cluster Diameter (Δ), low values are better, which means a higher compactness of the clusters. For Dunn Index (D), high values are better, which means compact and well-separated clusters, using the ratio between separation and diameter measures (δ/Δ). And for Davies-Bouldin Index (DB), lower values are better, which also means compact and well separated clusters, using the ratio among compactness and attribute value differences between clusters. The analysis of the algorithm G-means with 14 clusters was impossible to be created, due to automatic nature of determination of the number of clusters. For that reason, it is not described. The next four tables summarize the results obtained in the analysis (see Tables 3.10, 3.11, 3.12, 3.13). In each table, for each CVI, an arrow points to the direction (up, down) where the (high, low) values are the best ones for ensuring a "good" definition of the clusters, and the best values for each CVI across the three methods are outlined in boldface letter.

Table 3.7: Cluster Validation Indexes tested (δ, Δ, D, DB) with 2 clusters.

	(δ)↑	(Δ)↓	(D)↑	(DB)↓
K-means, 2 clusters	**3.07**	9.97	**0.30**	**3.50**
G-means, 2 clusters	2.96	9.96	0.29	3.53
Hier. Agglom., 2 clusters	2.93	**9.78**	0.29	3.85

Table 3.8: Cluster Validation Indexes tested (δ, Δ, *D*, *DB*) with 5 clusters.

	(δ)↑	(Δ)↓	(*D*)↑	(*DB*)↓
K-means, 5 clusters	3.64	**12.05**	**0.30**	4.25
G-means, 5 clusters	**3.66**	**12.05**	**0.30**	4.51
Hier. Agglom., 5 clusters	3.07	12.75	0.24	**4.24**

Table 3.9: Cluster Validation Indexes tested (δ, Δ, *D*, *DB*) with 9 clusters.

	(δ)↑	(Δ)↓	(*D*)↑	(*DB*)↓
K-means, 9 clusters	4.17	**12.30**	**0.34**	4.73
G-means, 9 clusters	**4.23**	**12.30**	**0.34**	**4.62**
Hier. Agglom., 9 clusters	3.20	16.08	0.20	4.96

Table 3.10: Cluster Validation Indexes tested (δ, Δ, *D*, *DB*) with 14 clusters.

	(δ)↑	(Δ)↓	(*D*)↑	(*DB*)↓
K-means, 14 clusters	**4.56**	**13.80**	**0.33**	**4.65**
Hier. Agglom., 14 clusters	3.56	18.81	0.18	7.51

Based on the examination of the values of the different indexes for the different clustering techniques and of the number of clusters, it was discovered that K-means with 9 clusters and G-means with 9 clusters were the ones with the best values. Over a global analysis of the indexes, it could be detected that both algorithms were the ones with higher Dunn Index (*D*) value, meaning that its relationship between compactness and separation was the better one. However, G-means was a little better, because its Davies-Bouldin Index (*DB*), which is also measuring the relationship between compactness and separation, was lower.

Another point that reinforces and complements the statement that clusters' composition and the number of clusters obtained were meaningful, was the validation analysis of the experts' domain (the client), which will be detailed in the next subsection. With all the analyses done, it was decided to use the G-means algorithm with 9 clusters as the definitive clustering technique.

3.4.3.2 User Validation through the Interpretation of the Clusters

After the ending of both kind of analysis, sensitivity analysis and structural validation analysis, it could be stated that the selection of G-means clustering technique, especifically with 9 clusters, was a good selection, and the obtained 9 clusters were structurally well defined.

To end with the post-processing and validation of the descriptive model obtained (the set of clusters), The set of 9 clusters obtained should be interpreted by the end users (customers).

A common tool, which helps in this interpretation process, is the computation of the *centroids* or *barycentres* of each cluster. The centroid of one cluster is a possibly virtual object which is the average prototype of the objects in the cluster. Thus, the numerical variables are averaged to get a mean value for those variables, and the mode of qualitative variables are used as the most frequent value of those variables.
In this market basket case, the objects of the database, and the clusters were described with near 400 variables (395). Thus, the number of variables was very high to provide insightful meaning to the experts. This high number of variables could not give a good summary of the profile of each cluster. In the same way, the centroids of each cluster should have the same number of variables.

Hence, both to better interpret the clusters and the centroids, the number of variables must be reduced. Two ways of reduction of the variables could be applied:

- The use of a feature selection strategy, based on the *detection of the most important or relevant variables*, and use only the most important ones to describe both the clusters and the centroids.
- *Aggregation or generalization of the variables using the temporal relations* among several variables to reduce the number of variables.

In the next subsections, the application of these strategies will be explained.

3.4.3.2.1 Computing the Relevance of the Variables

Once selected the final clustering, and aiming at detecting the most relevant variables for the characterization of the clusters, the cluster label obtained with the clustering process, was added as a new class variable to the whole dataset. Then, a discriminant model was selected to get a classifier model, and at the same time, to get the set of the discriminant or predictive attributes. The used discriminant model, due to its good accuracy properties, was an ensemble strategy based on Random Forests. This way, in addition to the set of classifiers (the ensemble of Decision Trees), the set of relevant/discriminant attributes was obtained. The Random Forest was created with 100 trees. The top set of 16 most predictive features discovered were the following:

1. Numero medio unidades por ticket - Trimestre 2: 2.98%
2. Numero medio unidades por ticket - Trimestre 3: 2.69%
3. Coverage % PASTA: 2.49%
4. Mean Ticket - Trimestre 4: 2.27%
5. Mean Ticket - Trimestre 1: 1.99%
6. Coverage % FORMATGES ESPECIALIT.REGIONALS: 1.89%
7. % Unidades del total es venen en Dimecres - Trimestre 1: 1.88%
8. Mean Ticket - Trimestre 3: 1.64%
9. Numero medio unidades por ticket - Trimestre 1: 1.52%
10. Mean Ticket - Trimestre 2: 1.47%
11. Numero medio unidades por ticket - Trimestre 4: 1.41%
12. Coverage % AIGUES: 1.37%
13. Coverage % CAFES I SUCCEDANIS: 1.36%
14. Coverage % BEGUDES ESPUMOSES: 1.33%
15. Coverage % PLATS CUINATS DE CARN: 1.30%
16. Region: 1.29%

This huge reduction of attributes, from 395 to 16, was communicated to the end users, for their validation. Even though they agreed on the set of relevant attributes obtained as being meaningful, they outlined a new problem: the description of the clusters and the centroids should be done through variables with general information, and not expressing partial information of one period of time (quarters). For instance, from the point of view of the client describing the clusters and/or the centroids using the *% Unidades del total es venen en Dimecres - Trimestre 1: 1.88%* attribute hadn't value because it was too concrete, and was not useful for marketing purposes too. The end users wanted more general information in the descriptions.

3.4.3.2.2 Generalization of the Variables Using Temporal Relations

Aiming at solving this problem of the appearance of variables referring to a concrete period of time, the solution undertaken was to make an aggregation process over the same variables during the four quarters of the year.

To do that, based on the set of most predictive features obtained in the previous feature selection step, the features used in the cluster description and/or centroid description where the overall abstract concept that represented the original feature. For instance, the feature *Numero medio unidades por ticket - Trimestre 2: 2.98%* was aggregated with all the other variables referring to the same concept in all the remaining quarters of the year. It was named as *Unidades/ticket* and the value of it was computed as the aggregation in order to capture the behaviour through all the year.

In the Figure 3.4 there is a scheme of this aggregation process over the variables with temporal relationships.

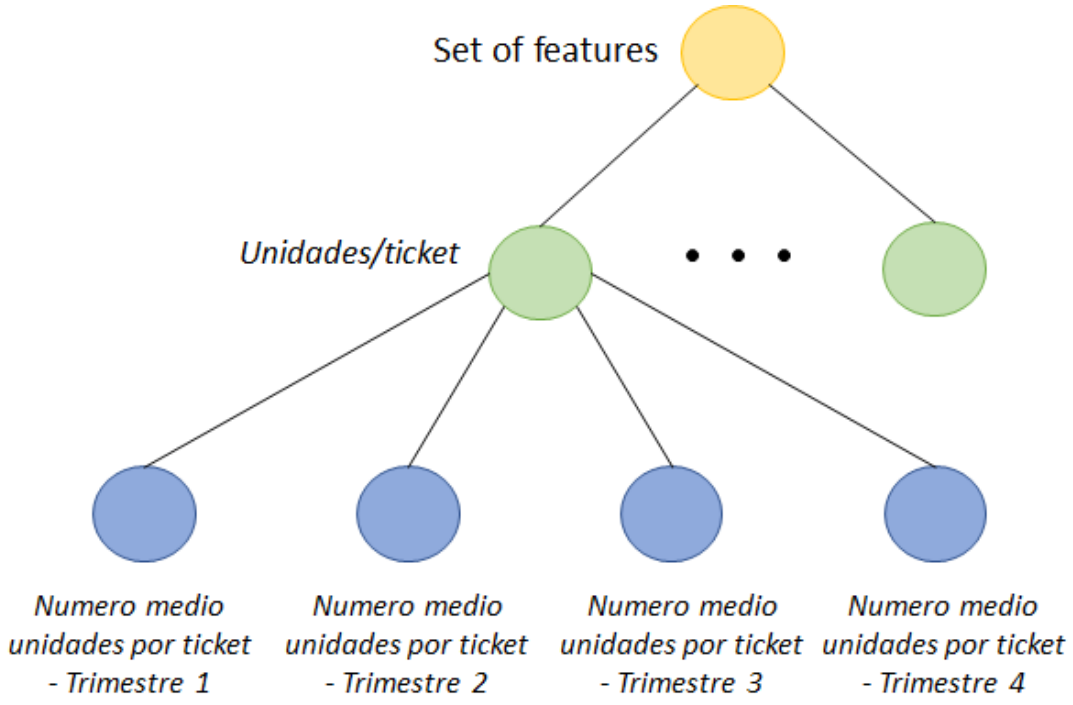


Figure 3.4 Aggregation process over the variables with temporal relationship

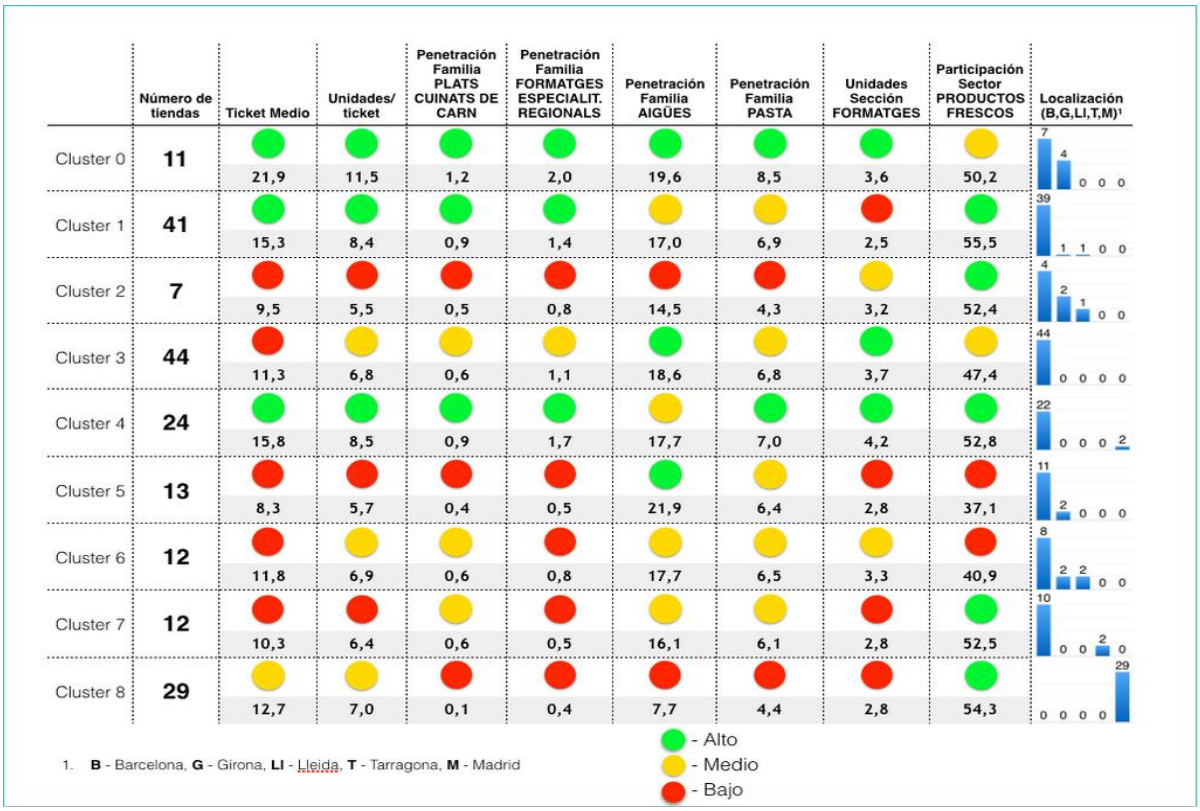
After these aggregations, the set of features finally used to describe the clusters and the centroids was the following:

1. Numero de tiendas
2. Ticket Medio
3. Unidades/ticket
4. Penetración Familia PLATS CUINATS DE CARN
5. Penetración Familia FORMATGES ESPECIALIT. REGIONALS
6. Penetreación Familia AIGÜES
7. Penetreación Familia PASTA
8. Unidades Sección FORMATGES
9. Penetración Sector PRODUCTOS FRESCOS
10. Participación Sector PRODUCTOS FRESCOS
11. Localización

The next tables were created in order to visualize the differences between clusters (Tables 3.11, 3.12). The colour of each variable is used to distinguish if the value was high (green), medium (orange) or low (red). Both tables are basically the same, the only difference is that on the second table the value of the feature is added.

Table 3.11: Cluster feature summary.

Table 3.12: Cluster feature summary with values.



	Número de tiendas	Ticket Medio	Unidades/ ticket	Penetración Familia PLATS CUINATS DE CARN	Penetración Familia FORMATGES ESPECIALIT. REGIONALS	Penetración Familia AIGÜES	Penetración Familia PASTA	Unidades Sección FORMATGES	Participación Sector PRODUCTOS FRESCOS	Localización (B,G,LI,T,M)[1]
Cluster 0	11	21,9	11,5	1,2	2,0	19,6	8,5	3,6	50,2	7 4 0 0 0
Cluster 1	41	15,3	8,4	0,9	1,4	17,0	6,9	2,5	55,5	39 1 1 0 0
Cluster 2	7	9,5	5,5	0,5	0,8	14,5	4,3	3,2	52,4	4 2 1 0 0
Cluster 3	44	11,3	6,8	0,6	1,1	18,6	6,8	3,7	47,4	44 0 0 0 0
Cluster 4	24	15,8	8,5	0,9	1,7	17,7	7,0	4,2	52,8	22 0 0 0 2
Cluster 5	13	8,3	5,7	0,4	0,5	21,9	6,4	2,8	37,1	11 2 0 0 0
Cluster 6	12	11,8	6,9	0,6	0,8	17,7	6,5	3,3	40,9	8 2 2 0 0
Cluster 7	12	10,3	6,4	0,6	0,5	16,1	6,1	2,8	52,5	10 0 0 2 0
Cluster 8	29	12,7	7,0	0,1	0,4	7,7	4,4	2,8	54,3	0 0 0 0 29

1. **B** - Barcelona, **G** - Girona, **LI** - Lleida, **T** - Tarragona, **M** - Madrid

- Alto
- Medio
- Bajo

3.4.4 Association discovery

The last part of the project was the discovering association rules for each cluster. At the beginning of the project, it was defined to obtain association rules for each cluster. However, at the end, the client decided to start with the associations rules of the cluster with more instances. With that, the client could analyse whether the results obtained had sense.

The cluster chosen to analyse its associations was the number 4. This cluster was composed by 44 stores, and the nearest store to its centroid (i.e., the medioid) was the store with id *605*. The next figure is a list of the stores that composed the cluster (Figure 3.5).

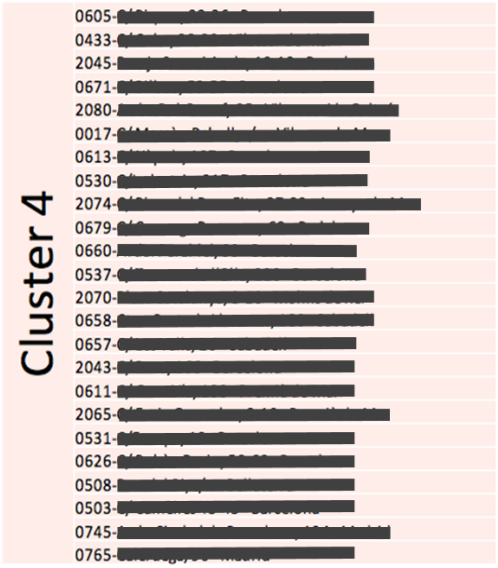


Figure 3.5: List of stores in cluster 4.

To analyse the association rules of the store, a new dataset was created where each instance was a ticket, and the features were the set of items purchased that composed the ticket (Figure 3.6).

Trans-id	Products
12345	product_A, product_B, product_C, product_D
67890	product_A, product_E
67890	product_B, product_C, product_F

Figure 3.6: Structure of the new dataset.

Through the process of association discovery, two versions of association rules were created. In the first one, items in sale were added on the association discovery. In the second one, items on sale were removed. The hypothesis back this reasoning is that items in sale can create false associations because its temporal condition. For instance, a product that has a special offer will be purchased more often than usual.

After the analysis of both resulting set of rules (one with items in sale and the other without), it was decided to deliver the results corresponding to the second version, the one without products in offer. From now on, all the results and processes performed in this memory are the ones corresponding to this version.

Antecedent	Consequent	Coverage	Support	Confidence	Leverage	Lift
VERDURES I HORTALISSES	FRUITES	30.2080%	13.5530%	44.8650%	6.4890%	1.9185
FRUITES	VERDURES I HORTALISSES	23.3860%	13.5530%	57.9540%	6.4890%	1.9185
FRUITES	IOGURTS	23.3860%	5.8940%	25.2030%	2.3540%	1.6651
IOGURTS	FRUITES	15.1360%	5.8940%	38.9400%	2.3540%	1.6651
LLET LIQUIDA	VERDURES I HORTALISSES	15.7400%	7.0280%	44.6520%	2.2730%	1.4782
VERDURES I HORTALISSES	LLET LIQUIDA	30.2080%	7.0280%	23.2650%	2.2730%	1.4782
VERDURES I HORTALISSES	IOGURTS	30.2080%	6.8290%	22.6070%	2.2570%	1.4936
IOGURTS	VERDURES I HORTALISSES	15.1360%	6.8290%	45.1190%	2.2570%	1.4936
FRUITES	LLET LIQUIDA	23.3860%	5.8910%	25.1910%	2.2100%	1.6005
LLET LIQUIDA	FRUITES	15.7400%	5.8910%	37.4290%	2.2100%	1.6005
OUS	VERDURES I HORTALISSES	8.1780%	4.6380%	56.7090%	2.1670%	1.8773

Figure 3.7: Associations rules ordered according to Leverage measure.

The measures used to filter/order associations rules were Lift and Leverage. Both ordering strategies were used, and for each of them, it was obtained a set of rules where each one maximizes the measure selected. In addition, rules were created with one antecedent by demand of the client, because he was interested in simple rules that could easily apply. The next figures are some of the rules discovered via Leverage and Lift measures ordering (Figures 3.7, 3.8).

Antecedent	Consequent	Coverage	Support	Confidence	Leverage	Lift
PARAMENT DE MA	LLAR	0.2800%	0.0030%	0.9120%	0.0020%	34.7925
LLAR	PARAMENT DE MA	0.0260%	0.0030%	9.7560%	0.0020%	34.7925
BOLQUERS	PUERICULTURA	0.4380%	0.0090%	2.0420%	0.0090%	23.3119
PUERICULTURA	BOLQUERS	0.0880%	0.0090%	10.2190%	0.0090%	23.3119
CUINA	LLAR	0.4910%	0.0030%	0.5860%	0.0030%	22.3629
LLAR	CUINA	0.0260%	0.0030%	10.9760%	0.0030%	22.3629
PARAMENT DE MA	CUINA	0.2800%	0.0270%	9.4640%	0.0250%	19.2832
CUINA	PARAMENT DE MA	0.4910%	0.0270%	5.4070%	0.0250%	19.2832
PLATS CUINATS DE VEGETALS	PLATS CUINATS DE PEIX	0.3900%	0.0120%	3.0300%	0.0110%	17.6162
PLATS CUINATS DE PEIX	PLATS CUINATS DE VEGETALS	0.1720%	0.0120%	6.8770%	0.0110%	17.6162
BOLQUERS	ALIMENTS INFANTILS	0.4380%	0.0790%	18.0890%	0.0750%	17.2589

Figure 3.8: Associations rules ordered according to Lift measure.

Figure 3.9: CleverData web to visualize the results.

3.4.5 Results Delivery

To provide to the client an agile and dynamic way to analyse the results, it was created a web page where the results of the project were published (Figure 3.9). With that, the client could access the results wherever he/she needed.

The web was divided in 4 pages. Each of them described a specific part of the project. In addition, three files were delivered. The first one, was an excel file with information of the different clusters and its stores' composition. The last two files were the list of associations rules obtained using the lift and leverage measures for ordering.
The first page of the web page was a descriptive analysis of the stores that composed each cluster and its characteristics. In addition, for each cluster, it was created a map where the position of the stores that composed the cluster was plotted. In addition, stores from its main competitor were plotted as well. With that, the client could analyse the distance between stores. These maps were created using the Carto software tool [Carto, 2017]. The next image is an example of the composition of the page corresponding to the cluster 4 (Figure 3.10).

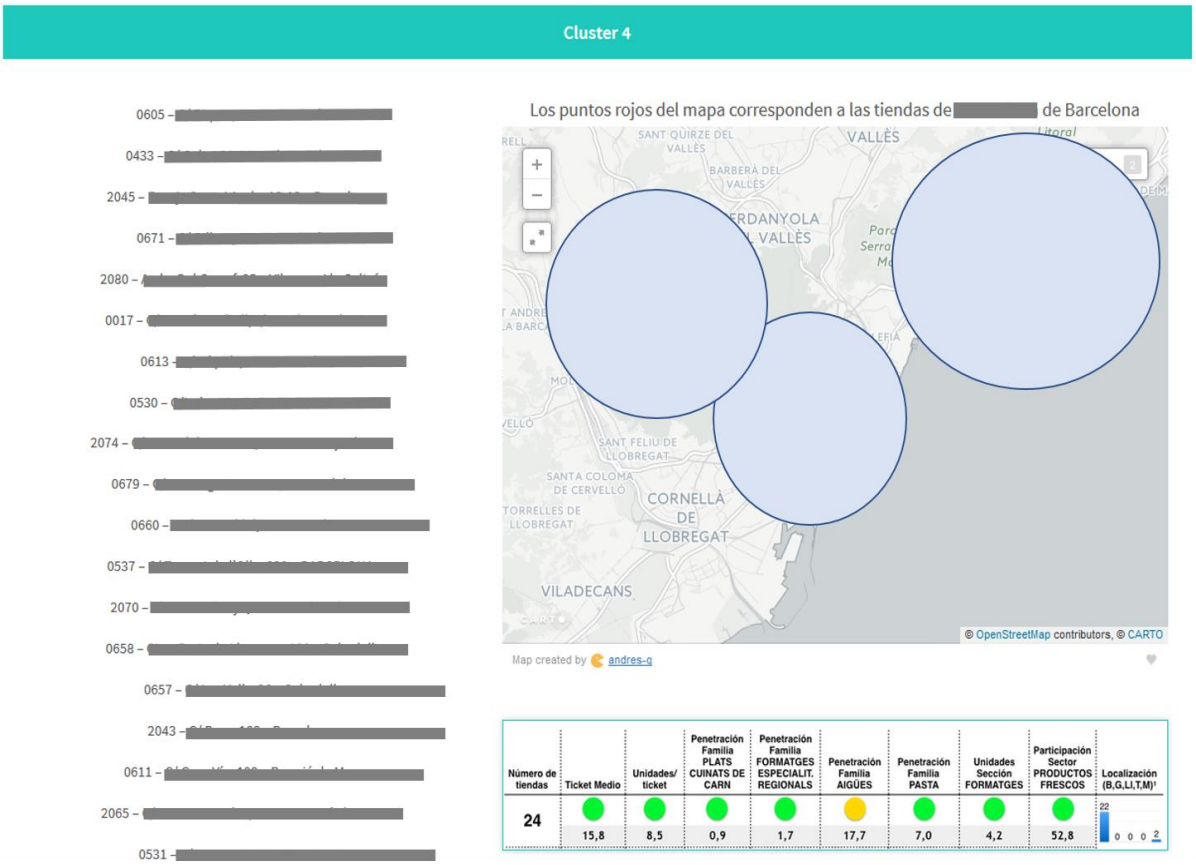


Figure 3.10: Cluster 4 visualization.

In the second page created, the association rules discovered using the Lift and Leverage measures were plotted. In addition, it was added two relationship diagrams for both strategies (Figures 3.11, 3.12). With them, the client could visualize how association rules were connected.

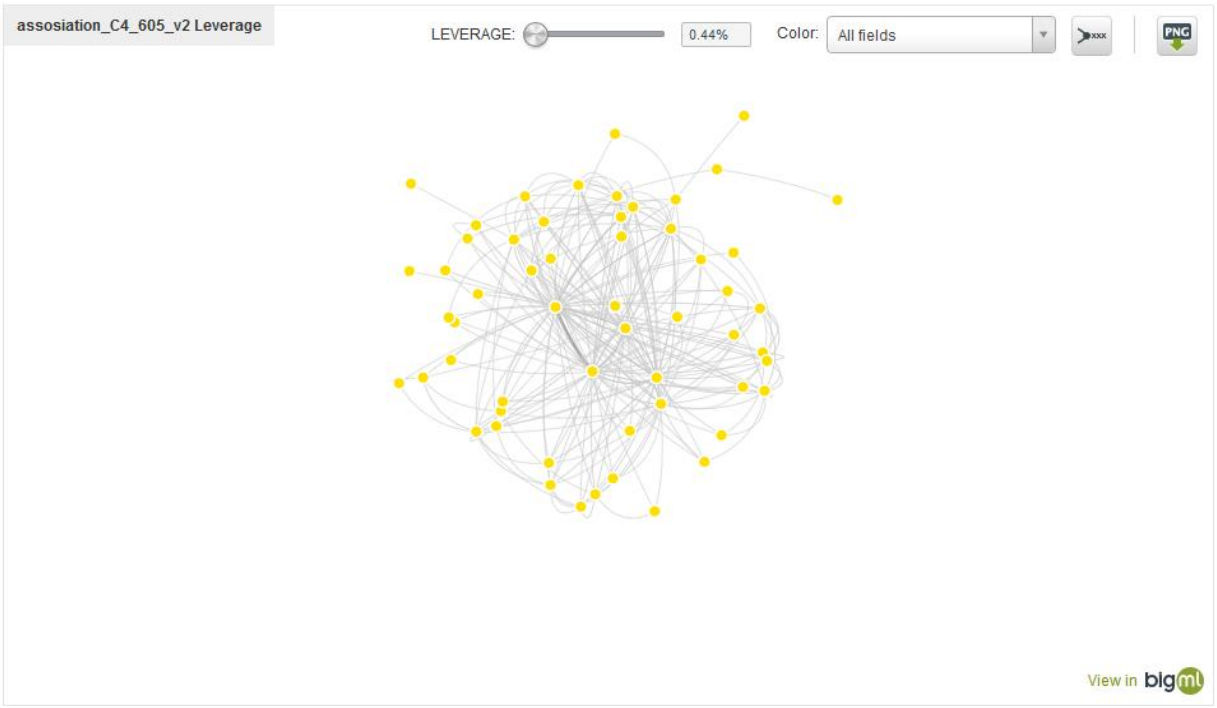


Figure 3.11: Leverage Diagram.

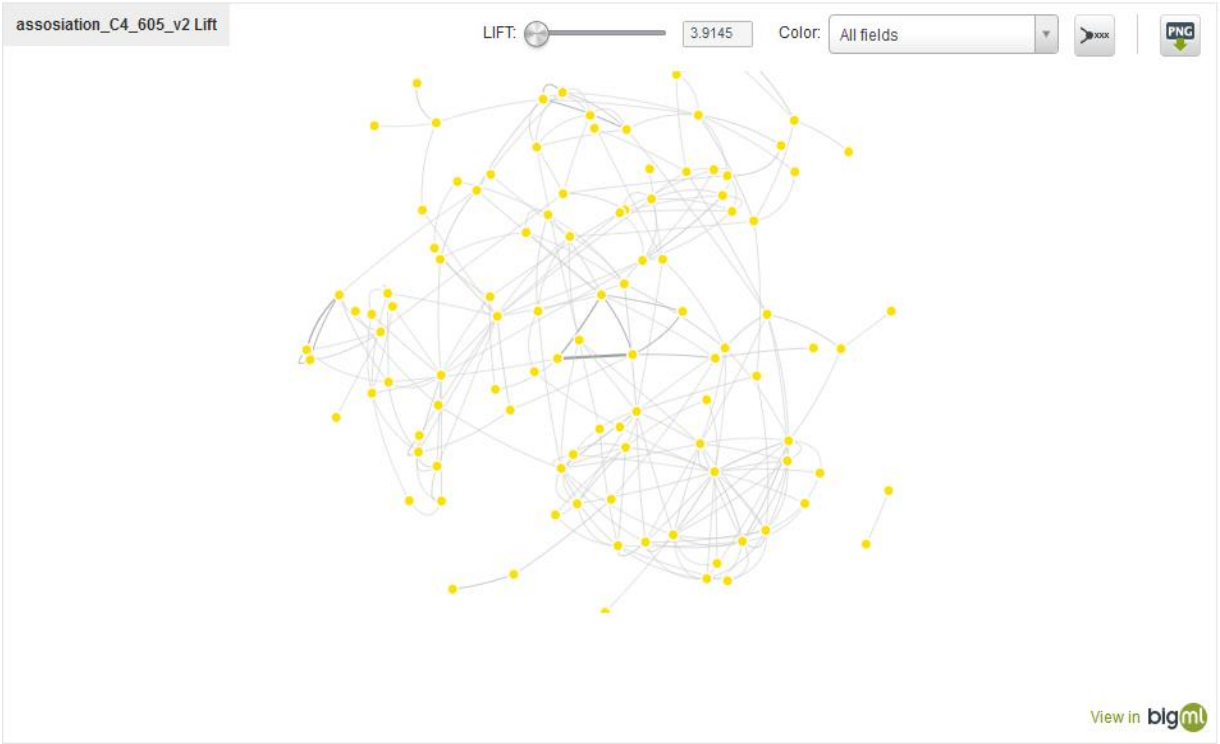


Figure 3.12: Lift Diagram.

The third page created was a reminder of which metrics have association rules, and how can they be interpreted (Figures 3.13, 3.14).

Familia 1	Familia 2	Coverage (penetración)	Support	Confidence	Leverage	Lift
PASTA	SOPES, BROU I PURES	6,4%	1,1%	17,8%	0,74%	2,9

- **Coverage (penetración)**: porcentaje de tickets con PASTA.
- **Support**: porcentaje de tickets con PASTA y SOPAS.
- **Confidence**: de las veces que se compra PASTA, qué porcentaje se compra también SOPAS.
- **Leverage**: comprar PASTA y SOPAS simultáneamente sucede un 0,74% más a menudo que si fueran estadísticamente independientes. Un "cero" indica que la compra simultánea de los 2 productos es aleatoria. Valores > 0 indican asociación positiva
- **Lift**: si se compra PASTA, es 2,9 veces más probable que se compre SOPAS. Un "uno" indica que no hay asociación. Cuanto mayor es el valor, mayor es la fuerza de la asociación. "Premia" las asociaciones con pocas ocurrencias.

Figure 3.13: Metrics.

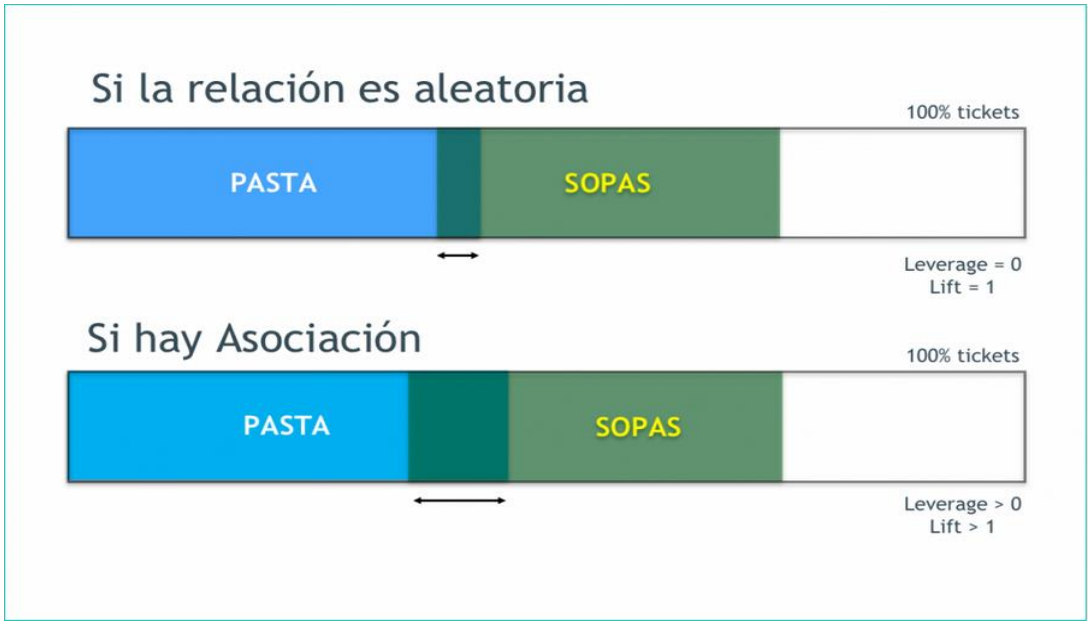


Figure 3.14: Description of rules.

The last page created was a dynamic scatterplot. With that, the client was able to plot different combinations of features in order to analyse the correlation between them. These features were the set of features used in the clustering dataset. Some examples of scatterplots were the following: Ticket mean price (Figure 3.15), Region (Figure 3.16) and Mean units per ticket (Figure 3.17). For the three scatterplots, X-axis correspond to the set of clusters, and Y-axis correspond the value of the feature analysed. Points in the scatterplot are stores.

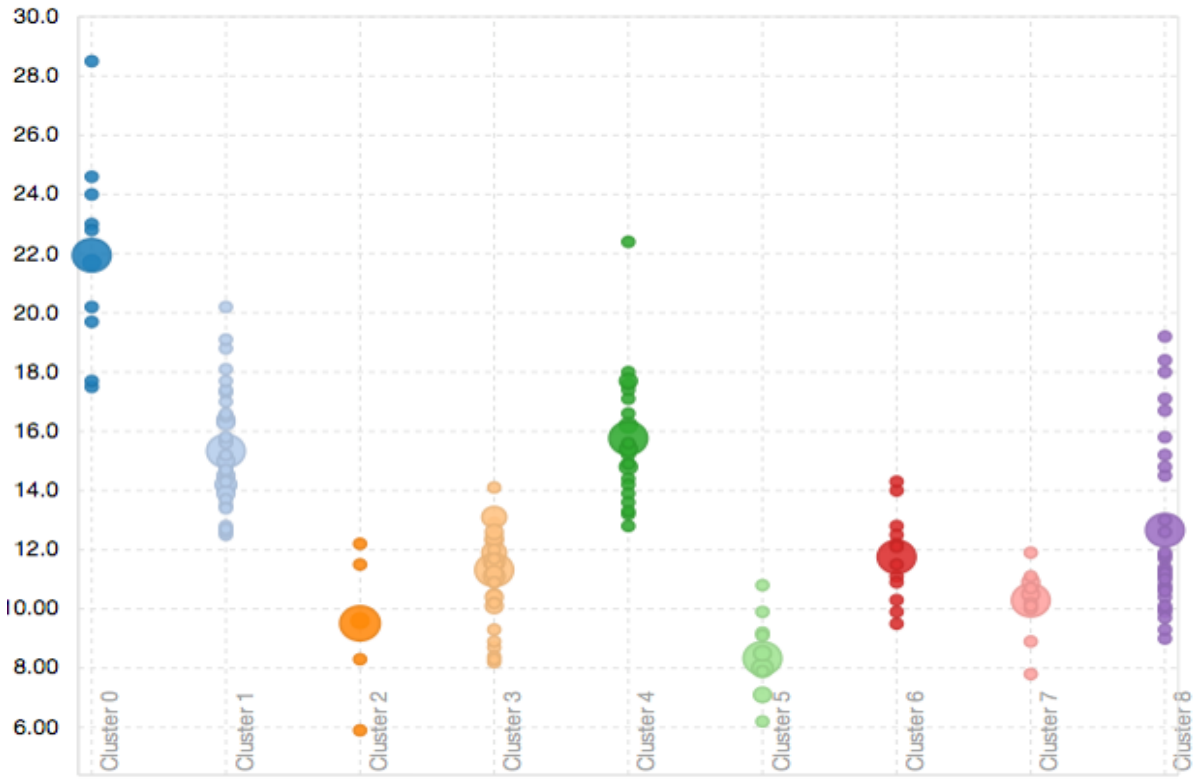


Figure 3.15: Mean price ticket scatterplot.

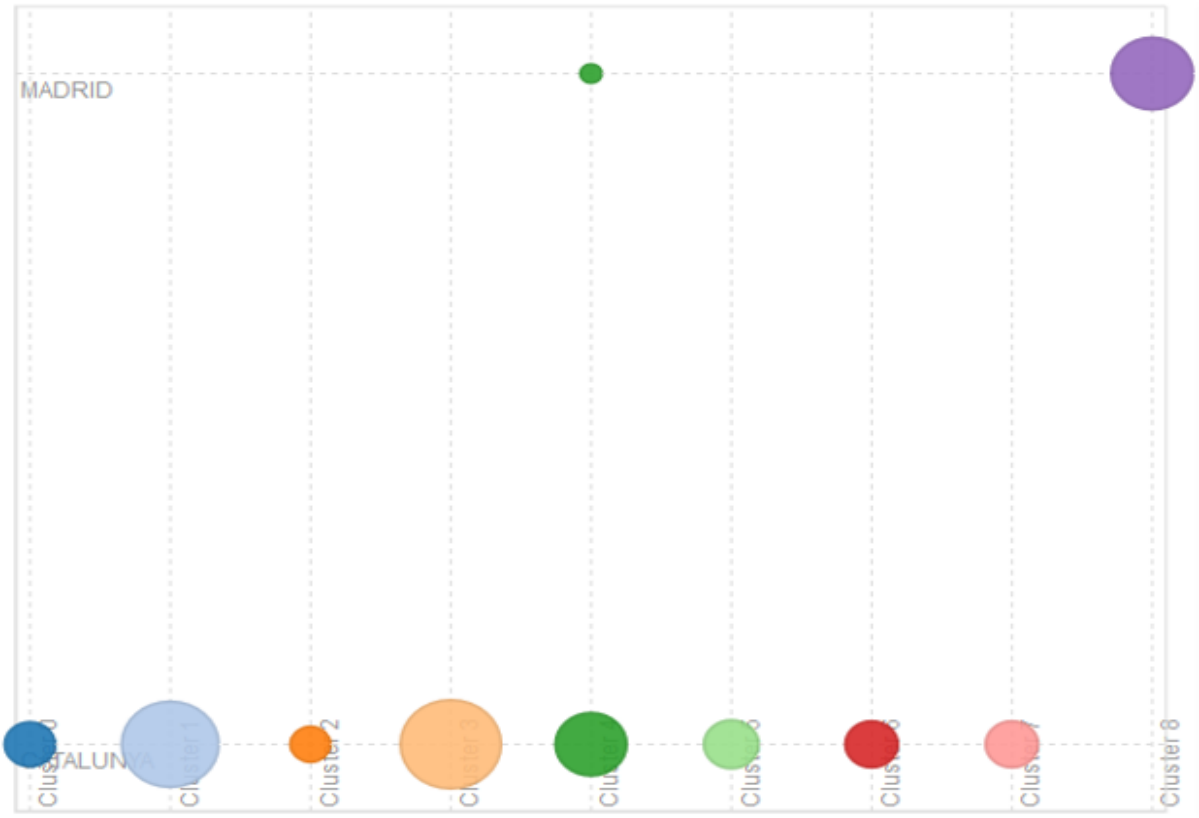


Figure 3.16: Region scatterplot.

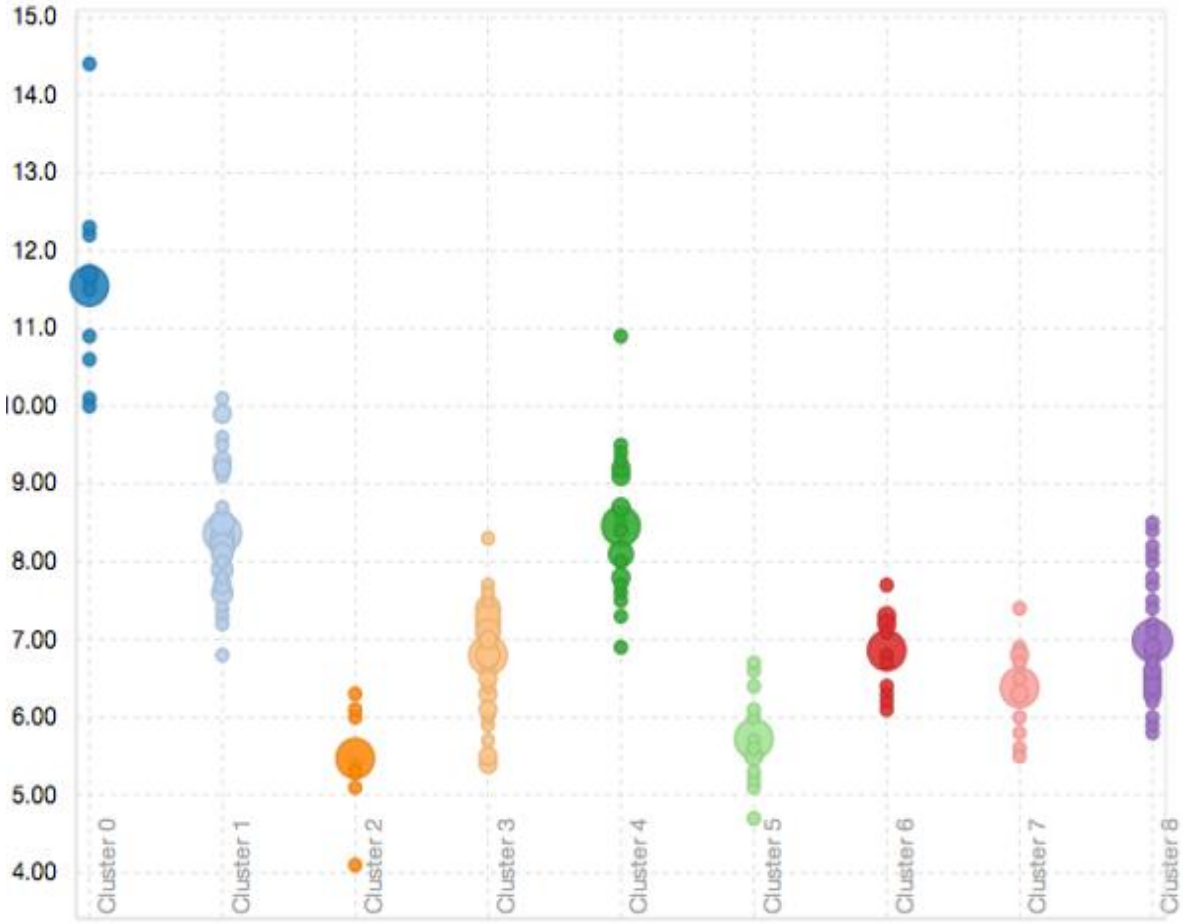


Figure 3.17: Mean units per ticket scatterplot.

Chapter 4

Conclusions

In this Master Thesis report, a Market Basket Analysis project in Retail was described. Through the project, an analysis of several clustering algorithms was performed. Results of the study showed that the composition of the clusters using K-means, G-means and Hierarchical agglomerative algorithms was similar. In addition, based on the Cluster Validation Indexes, the structural analysis study showed that G-means with 9 clusters was a good choice for implementing the clustering process.

Cluster validation through the interpretation of the clusters provided a simple way to understand clusters. Computing the relevance of the variables, using a Random Forest, allowed to obtain a set of the most representing features that defined the clusters. However, the generalization of the variables using temporal relations was needed in order to obtain a more abstract representation of the features of the clusters, which provided valuable results to the client.

For one of the clusters, a study of the most commonly purchased items was performed. Using Lift and Leverage measures as filtering and ordering techniques, association rules were discovered. Moreover, a web page was created to provide an agile way to provide and analyse the results.

Results obtained were satisfactory according to the feedback of the client. The objectives defined at the beginning of this study were achieved:

- *The analysis of the possibly different selling behaviour of the stores/shops of the client*
- *The analysis of customers' purchase behaviour*

The *knowledge and patterns discovered* by means of the use of several data mining methods such as descriptive models (clustering techniques), discriminant methods (Random Forest) used as feature selection technique, and associative models (association rules) would provide to the client an insightful analysis about its stores and customers behaviour and a competitive advantage over its competitors. In addition, the clustering analysis woull provide a new vision of store's behaviour that can lead to future strategies.

4.1 Difficulties between the scientific world and the company goals

A constraint encountered through all the project was the adaptation of the data scientist to the client's needs. Usually, scientific approaches can clash with a company goals and approaches. Sometimes the solutions provided are impossible to perform by the company, or results are too complex or the company desires a specific solution even if it is possible to achieve a better one. All these scenarios are common in data mining projects, and in general, in the interaction between the scientific/technological world and the economic/company world.

In this project, several steps in the project were adapted to fit the special requirements of the client. For instance, the use of clusters was done exactly for this reason. From the scientific point of view, discovering association rules for each one of the stores was a very reasonable solution, taking into account that there was many data regarding the purchases on each store. There is no more reliable rules than the ones discovered using exclusively the tickets of that store. However, this solution was impracticable from the point of view of the company because it is impossible to create specific actions for each store.

Another task performed in this project which its solution was adapted due the client's needs was the interpretation of the clusters. The interpretation of the clusters step was created in order to provide the client a simple way to understand the clusters. From the point of view of the company, characterise and identify the profiles in the clusters based on the huge number of features used in the clustering was impossible. In addition, the agglomeration of quarterly features was performed because from the point of view of the client, it did not provide a good representation of the clusters.

However, even with all these constraints, data mining projects have to be performed. Our recommendation is to have always in mind that results must be meaningful and applicable. Even if the project is perfect, it would have no value if the client cannot understand and/or apply the results, because at the end, the client is the one that make the corresponding actions based on the results. In our opinion, we would say that the key of a project success is the balance of the requirements and needs of both worlds.

4.2 Future Work

At the end of the project, an exhaustive analysis of the decisions made through the project was performed. The aim of this analysis was the detection of aspects of the project that could be improved or hypotheses that could be interesting to study.

One aspect that could be improved is related to the step of association rules discovering. In this project, in order to obtain the association rules in each cluster, it was selected the nearest store to the centroid (i.e., the medioid). Thus, using the tickets record of that store, association rules were discovered, and the resulting set of rules were extrapolated to all the

stores that composed that cluster. This approach was performed because for the client, it was easier to understand the resulting co-occurrences and behaviour patterns of just one store. However, through this process, valuable association rules and generalization can be lost because the association rules discovered are just from one store. To improve this approach, a reasonable solution is discovering the association rules using the historical tickets of the entire set of stores composing the cluster. This solution has an increment in terms of computational cost and feature engineering, but solves the possible lack of generalization of the association rules.

A study that could be interesting to perform is related to the features used in the clustering dataset. Features were created to capture the behaviour of each quarter. This decision was made first, because consumers' behaviour change through quarters and second, because depending on the quarter, specific marketing actions are made. Therefore, stores clusters had to be created capturing the quarterly behaviour. However, it could be interesting to create those features in different intervals of time, like months or weeks, and analyse the resulting clusters to evaluate their quality.

To conclude, a new approach creating the clusters could be studied. The new approach would be defined as: once created the clusters using the set of features obtained through the feature engineering process, train a Random Forest (as it was described in the previous chapter) to obtain the most important features. Then, based on those features, run again a new clustering process just using the new set of relevant features. This way, irrelevant features are not considered, feature selection is performed and the complexity of the model decrease exponentially. The hypothesis of this approach is that new clusters obtained using this new approach will be more robust to noisy features that do not add valuable information to the clusters.

References

Bibliographic references

[Agrawal & Srikant, 1994] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB 94)*, pp. 487-499, Santiago, Chile, September 1994.

[Agrawal *et al.*, 1993] R. Agrawal, T. Imielinski, and A. Swami. Mining associations between sets of items in large databases. In *Proc. of the ACM SIGMOD Int'l Conference on Management of Data*, pp. 207-216, Washington D.C., May 1993.

[Anderson & Darling, 1954] Anderson, T.W.; Darling, D.A. A Test of Goodness-of-Fit. *Journal of the American Statistical Association* 49:765-769, 1954. [DOI: 10.2307/2281537].

[Bastide et al., 2000] Y. Bastide, Pasquier N, Taouil R, Stumme G and Lakhal L. Mining minimal non-redundant association rules using frequent closed itemsets. In: *1st International Conference on Computational Logic (CL 2000)*, pp. 972-986. Springer-Verlag: Berlin, 2000.

[Breiman, 2001] L. Breiman. Random forests. *Machine Learning* 45(1):5–32, 2001.

[Breiman, 1996] L. Breiman. Bagging predictors. *Machine Learning* 24(2):123–140, 1996.

[Breiman *et al*., 1984] L. Breiman, J.H. Friedman, R.A. Olshen and C.J. Stone. Classification and regression trees. Monterey, CA: Wadsworth & Brooks/Cole Advanced Books & Software, 1984.

[Brin *et al.*, 1997] S. Brin, R. Motwani, J. D. Ullman, and S. Tsur. Dynamic itemset counting and implication rules for market basket data. In *Proc. of the ACM SIGMOD Int. Conference on Management of Data (ACM SIGMOD '97)*, pp. 265-276, 1997.

[Brun *et al*., 2007] M. Brun, C. Sima, J. Hua, J. Lowey, B. Carroll, E. Suh, and E.R Dougherty. Model-based evaluation of clustering validation measures. *Pattern Recognition*, 40(3):807-824, 2007.

[Dal Pozzolo & Bontempi, 2015] A. Dal Pozzolo & G. Bontem*pi. Adaptive Machine Learning for Credit Card Fraud Detection. Ph.D. thesis in Computer Science.* 2015. [Available at: http://www.ulb.ac.be/di/map/adalpozz/pdf/Dalpozzolo2015PhD.pdf].

[Davies & Bouldin, 1979] David L Davies and Donald W Bouldin. Cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(2):95–104, 1979.

[Defays, 1977] D. Defays. An efficient algorithm for a complete-link method. *The Computer Journal*. British Computer Society 20(4): 364–366, 1977. [DOI:10.1093/comjnl/20.4.364]

[Dunn, 1974] Joseph C Dunn. Well separated clusters and optimal fuzzy partitions. *Journal of cybernetics*, 4(1):95–104, 1974.

[Everitt, 2011] Brian Everitt. *Cluster Analysis*. Chichester, West Sussex, United Kingdom:Wiley. ISBN 9780470749913, 2011.

[Freund & Schapire, 1997] Y. Freund and R. Schapire. A decision-theoretic generalization of online learning and application to boosting, *Journal of Computer and System Sciences*, Volume 55, Issue 1, August 1997.

[Freund 1995] Y. Freund, Boosting a weak learning algorithm by majority, *Information and Computation* 121, No. 2 (September 1995), 256-285; an extended abstract appeared in Proc. of the Third Annual Workshop on Computational Learning Theory, 1990.

[Halkidi *et al*., 2001] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. On clustering validation techniques. *Journal of Intelligent Information Systems*, 17(2):107-145, 2001.

[Hämäläinen, 2010] W. Hämäläinen. Efficient discovery of the top-k optimal dependency rules with the Fisher's exact test of significance. In *Proc. of the 10th IEEE International Conference on Data Mining*, pp. 196-205, 2010.

[Hamerly & Elkan, 2003]. G. Hamerly and C. Elkan. In Procc. of the 17th Annual Conference on Neural Information Processing Systems (NIPS), pages 281-288, December 2003.

[Han *et al.*, 2004] J, Han, J. Pei, Y. Yin and R. Mao. Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach. *Data Mining and Knowledge Discovery* 8:53-87, 2004.

[Han *et al.*, 2000] J, Han, J. Pei and Y. Yin. Mining Frequent Patterns without candidate Generation. In Proc. of ACM-SIGMOD *Int. Conference on management of Data (SIGMOD'00)*, pp. 1-12, Dallas, TX, USA, 2000.

[Hennig and Liao, 2010] Christian Hennig and Tim F Liao. Comparing latent class and dissimilarity based clustering for mixed type variables with application to social stratification. Technical report, 2010.

[Jain & Dubes, 1988]. A. K. Jain and R.C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall: New Jersey, USA, 1988.

[Johnson, 1967] S. C. Johnson. Hierarchical Clustering Schemes. *Psychometrika*, 2:241-254, 1967.

[Kamakura, 2012] W. A. Kamakura. Sequential market basket analysis. *Mark Lett* 23:505–516, 2012. [DOI: 10.1007/s11002-012-9181-6].

[Kaufman & Roussew, 1990] Kaufman, L., & Roussew, P. J. *Finding Groups in Data – An Introduction to Cluster Analysis*. A Wiley-Science Publication John Wiley & Sons, 1990.

[Lu & Fu, 1978] Shin-Yee Lu; King Sun Fu. A Sentence-to-Sentence Clustering Procedure for Pattern Analysis. *IEEE Transactions on Systems, Man, and Cybernetics* 8(5):381-389, 1978.

[Macnaughton Smith *et al*., 1965] Macnaughton Smith, P., Williams, W., Dale, M. and Mockett, L. Dissimilarity analysis: a new technique of hierarchical subdivision. *Nature* 202: 1034–1035, 1965.

[MacQueen, 1967] J. B. MacQueen. Some Methods for classification and Analysis of Multivariate Observations. *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, University of California Press, 1:281-297, 1967.

[Parker, 2011] He D., Parker D.S. (2011) Learning the Funding Momentum of Research Projects. In: Huang J.Z., Cao L., Srivastava J. (eds) *Advances in Knowledge Discovery and Data Mining (PAKDD 2011).* Lecture Notes in Computer Science, vol 6635. Springer, Berlin, Heidelberg

[Piatetsky-Shapiro, 1991] G. Piatetsky-Shapiro. Discovery, analysis, and presentation of strong rules. *Knowledge Discovery in Databases*, pp. 229-248, 1991.

[Pietracaprina *et al.*, 2010] A. Pietracaprina, M. Riondato, E. Upfal and F. Vandin. Mining top-k frequent itemsets through progressive sampling. *Data Mining and Knowledge Discovery* 21(2):310-326, 2010.

[Pollack, 2016] J. Pollack. Retail Clustering Methods. The Parker Avery Group. 2016. [Available at: http://www.parkeravery.com/pov_Retail_Clustering_Methods.html].

[Portugal *et al*., 2015] I. Portugal & P. Alencar & D. Cowan. The Use of Machine Learning Algorithms in Recommender Systems: A Systematic Review. arXiv.org. 2015. [Available at: http://researcher.ibm.com/view_pic.php?id=144, 2017.]

[Quinlan, 1993] J.R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, 1993.

[Quinlan, 1986] Quinlan, J. R. Induction of Decision Trees. *Machine Learning* 1:81-106, Kluwer Academic Publishers, 1986.

[Quinlan, 1983] J. R. Quinlan. Learning efficient classification procedures, In *Machine Learning: an artificial intelligence approach*, Michalski, Carbonell & Mitchell (eds.), pp. 463-482 Morgan Kaufmann, 1983. [DOI: 10.1007/978-3-662-12405-5_15].

[R. Kelley & Ming-Long, 2005] P. R. Kelley & L. Ming-Long. Spatial Distribution of Retail Sales. Proc. of *10th Annual Conference of Pacific Rim Real Estate Society*. 2005.

[Rokach & Maimon, 2005] L. Rokach and O. Maimon. Clustering methods. In *Data Mining and Knowledge Discovery Handbook*, pp: 321-352, Springer US, 2005.

[Sevilla-Villanueva *et al.*, 2016] B. Sevilla-Villanueva, K. Gibert and M. Sànchez-Marrè. Using CVI for Understanding Class Topology in Unsupervised Scenarios. Procc. of *17th Conference of the Spanish Association for Artificial Intelligence (CAEPIA 2016)*. Lecture Notes in Artificial Intelligence, Vol. 9868, pp. 135-149. Springer-Verlag, 2016.

[Sibson, 1973] R. Sibson. SLINK: an optimally efficient algorithm for the single-link cluster method. *The Computer Journal*. British Computer Society, 16 (1): 30–34, 1973. [DOI:10.1093/comjnl/16.1.30].

[Steinbach *et al.*, 2000] Steinbach, M., Karypis, G., & Kumar, V. A comparison of document clustering techniques. In *Procc. of KDD Workshop on Text Mining*, Vol. 400, No. 1, pp. 525-526, 2000.

[Ward, 1963] Joe H. Ward. Hierarchical Grouping to Optimize an Objective Function. *Journal of the American Statistical Association* 58 (301): 236–244, 1963. [DOI: 10.2307/2282967Joe Ward].

[Webb, 2011] G. I. Webb. Filtered-top-k Association Discovery. *Data Mining and Knowledge Discovery* 1(3):183-192, 2011.

[Webb, 2007] G.I. Webb. Discovering significant patterns. *Machine Learning*, p. 1-33, 2007.

[Webb, 2006] G.I. Webb. Discovering significant rules. In *Proc. of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2006)*, pp. 434-443, 2006.

[Zaki, 2004] M.J. Zaki .Mining non-redundant association rules. *Data Mining and Knowledge Discovery*, pp. 223-248, 2004.

[Zaki, 2000] M. J. Zaki. Scalable algorithms for association mining. *IEEE Transactions on Knowledge and Data Engineering* 12(3): 372–390, 2000. [DOI: 10.1109/69.846291]

[Zaki *et al*., 1997] M. J. Zaki, S. Parthasarathy, M. Ogihara and W. Li. Parallel Algorithms for Discovery of Association Rules. *Data Mining and Knowledge Discovery* 1:343-373, 1997.

Internet references

[Anomaly detection, 2017] Anomaly detection. Wikipedia. 2017. Available at: https://en.wikipedia.org/wiki/Anomaly_detection

[Association rule, 2017] Association rule learning. Wikipedia. Available at: https://en.wikipedia.org/wiki/Association_rule_learning, 2017

[BigML, 2017] BigML. 2017. Available at: https://bigml.com/

[Carto, 2017] Carto. 2017. Available at: https://carto.com/

[Database, 2017] Database. Wikipedia. 2017. Available at: https://en.wikipedia.org/wiki/Database

[Data cleaning, 2017] Data cleaning. Wikipedia. 2017. Available at: https://en.wikipedia.org/wiki/Data_cleansing

[Data warehouse, 2017] Data warehouse. Wikipedia. 2017. Available at: https://en.wikipedia.org/wiki/Data_warehouse

[ETL, 2017] ETL. Wikipedia. 2017. Available at: https://es.wikipedia.org/wiki/Extract,_transform_and_load

[Feature engineering, 2017] Feature engineering. Wikipedia. 2017. Available at: https://en.wikipedia.org/wiki/Feature_engineering

[Hadoop, 2017] Hadoop. 2017. Available at: hadoop.apache.org

[Jupyter-Notebook, 2017] Jupyter-Notebook. 2017. Available at: http://jupyter.org/

[Logistics, 2017] Logistics. Wikipedia. 2017. Available at: https://en.wikipedia.org/wiki/Logistics

[Market Penetration, 1999] Market Penetration. Inestopedia. 1999. Available at: http://www.investopedia.com/terms/m/market-penetration.asp

[Market Share, 1999] Market Share. Inestopedia.1999. Available at: http://www.investopedia.com/terms/m/marketshare.asp

[Pandas, 2017] Python. 2017. Available at: http://pandas.pydata.org/

[Pareto principle, 2017]Pareto principle. Wikipedia. 1999. Available at: https://en.wikipedia.org/wiki/Pareto_principle

[Python, 2017] Python. 2017. Available at: https://www.python.org/

[Scikit-learn, 2017] Scikit-learn. 2017. Available at: http://scikitlearn.org/stable/modules/clustering.html

[Spark, 2017] Spark. 2017. Available at: http://spark.apache.org/

Annexes

Annex A: Description of the variables in the three databases

In this annex, there are the lists of variables that composed each one of the databases used in the project: Tickets dataset, Items dataset and Stores dataset. Two columns compose these figures. The first one is the name of the variable. The second one is the type of the variable.

Feature	Type
COD_DIA	Date
COD_FRANJA_HORARIA	Categorical
COD_PUNTO_VENTA	Categorical
COD_ARTICULO	Categorical
COD_OFERTA	Categorical
COD_VALOR_ANIADIDO	Categorical
COD_MARCA_PROP	Categorical
COD_TIPO_LINEA	Categorical
COD_TICKET	Categorical
IMPORTE_PVP	Numerical
UNIDADES	Numerical
CANTIDAD	Numerical
IMPORTE_TICKET	Numerical

Figure A.1: Tickets dataset features.

Feature	Type
ARTICULO	Categorical
DEPARTAMENTO	Categorical
SECCION_VENTA	Categorical
VARIEDAD	Categorical
SUBFAMILIA	Categorical
FAMILIA	Categorical
SECCION	Categorical
SECTOR	Categorical
ESTRUCTURA	Categorical
SUBCATEGORIA	Categorical
CATEGORIA	Categorical
GESTOR	Categorical
PLANOGRAMA	Categorical
MARCA_PROPIA	Categorical
SEG_ALFABETICA	Categorical
GESTION_PIEZAS_PDV	Categorical
TOTAL	Categorical
COMPRADOR	Categorical
AGRUPACION	Categorical
JEFE_AREA_COMPRAS	Categorical
SECTOR_NEP	Categorical
SECCION_NEP	Categorical
OFICIO_NEP	Categorical
CATEGORIA_NEP	Categorical
FAMILIA_NEP	Categorical

SUBFAMILIA_NEP	Categorical
VARIEDAD_NEP	Categorical
PRODUCTO_APL	Categorical
PRODUCTO_ECO	Categorical
PRODUCTO_SGLU	Categorical
TIPO_ALTA	Categorical
NUEVA_MARCA	Categorical

Figure A.2: Items dataset features.

Feature	Type
PUNTO_VENTA	Categorical
TIP_PUNTO_VENTA	Categorical
ENSENA	Categorical
HISTORICO	Categorical
REGION	Categorical
PROVINCIA	Categorical
COMARCA	Categorical
MUNICIPIO	Categorical
DISTRITO	Categorical
COORD_ZONA	Categorical
SUPERVISOR	Categorical
TARIFA	Categorical
GAMA	Categorical
COMPARABLE_01	Categorical
COMPARABLE_02	Categorical
FECHA_APERTURA	Date
FECHA_CIERRE	Date
POSTAL	Categorical
EMPRESA	Categorical
ESTADO	Categorical
GRUPO_COMERCIAL	Categorical
GAMA_MINIMA	Categorical
AREA_METROPOLITANA	Categorical
BORRADO	Categorical
CLASIFICACION	Categorical

TARIFA_CESION	Categorical
PARKING	Categorical
GRUPO_CLIENTE	Categorical
TERCERO	Categorical
CEF	Categorical
COMPARABLE_CHARCUTERIA	Categorical
COMPARABLE_CARNE	Categorical
COMPARABLE_FRUTA	Categorical
COMPARABLE_PESCADO	Categorical
COMPARABLE_PANADERIA	Categorical
MOSTRADOR_CHARCUTERIA	Categorical
MOSTRADOR_CARNE	Categorical
MOSTRADOR_FRUTA	Categorical
MOSTRADOR_PESCADO	Categorical
MOSTRADOR_PANADERIA	Categorical
FEC_ENSENA	Date
TALLA_CENTRO	Categorical
CIERRA_MEDIODIA	Categorical
METROS_CUADRADOS	Numerical

Figure A.3: Stores dataset features.

Annex B: List of features obtained from Feature Engineering steps

In this annex, there are the lists of variables used in each of the five different feature engineering versions. These features captured information about the stores and its behaviour. In addition, some features were created to capture market metrics like *Penetration rate* and *Market share rate*, or *the Pareto Principle*.

1. *Tenda*
2. *Max Ticket - Trimestre X*
3. *Mean Ticket - Trimestre X*
4. *Numero medio unidades diarios - Trimestre X*
5. *Numero medio unidades semanales - Trimestre X*
6. *Numero medio unidades mensuales - Trimestre X*
7. *% Unidades del total es venen en Dilluns - Trimestre X*
8. *% Unidades del total es venen en Dimarts - Trimestre X*
9. *% Unidades del total es venen en Dimecres - Trimestre X*
10. *% Unidades del total es venen en Dijous - Trimestre X*
11. *% Unidades del total es venen en Divendres - Trimestre X*
12. *% Unidades del total es venen en Dissabte - Trimestre X*
13. *% Unidades del total es venen en Diumenge - Trimestre X*
14. *Unidades venuts marca client- Trimestre X*
15. *Unidades venuts marca propia - Trimestre X*
16. *Numero referencias - Trimestre X*
17. *Relacio unidades client vs No client- Trimestre X*
18. *Relacio unidades venuts vs numero referencies - Trimestre X*
19. *Numero medio unidades por ticket - Trimestre X*
20. *Numero medio referencias por ticket - Trimestre X*
21. *TOP 1 Referencia aparece en mas tickets - Trimestre X*
22. *TOP 2 Referencia aparece en mas tickets - Trimestre X*
23. *TOP 3 Referencia aparece en mas tickets - Trimestre X*
24. *TOP 1 Referencia se venden mas unidades - Trimestre X*
25. *TOP 2 Referencia se venden mas unidades - Trimestre X*
26. *TOP 3 Referencia se venden mas unidades - Trimestre X*
27. *% ALIMENTACIO SECA sobre total ventas - Trimestre X*
28. *% FLECA I PASTISSERIA sobre total ventas - Trimestre X*
29. *% FORMATGES sobre total ventas - Trimestre X*
30. *% CONSERVES sobre total ventas - Trimestre X*
31. *% DERIVATS LACTIS sobre total ventas - Trimestre X*
32. *% XARCUTERIA TRADICIONAL sobre total ventas - Trimestre X*
33. *% PEIXOS I MARISC sobre total ventas - Trimestre X*
34. *% CARNS sobre total ventas - Trimestre X*
35. *% PRODUCTES PROMOCIONALS sobre total ventas - Trimestre X*
36. *% LIQUIDS I BEGUDES sobre total ventas - Trimestre X*
37. *% LLETS I BATUTS sobre total ventas - Trimestre X*

38. % DROGUERIA sobre total ventas - Trimestre X
39. % MATERIAL TENDES sobre total ventas - Trimestre X
40. % BASSAR sobre total ventas - Trimestre X
41. % ARTICLES PUBLICITARIS sobre total ventas - Trimestre X
42. % PERFUMERIA sobre total ventas - Trimestre X
43. % PLATS CUINATS/REFRIGERATS sobre total ventas - Trimestre X
44. % CONGELATS sobre total ventas - Trimestre X
45. % Altres seccions sobre total ventas - Trimestre X
46. % FRUITES I HORTALICES sobre total ventas - Trimestre X
47. % UNITAT CARNICA sobre total ventas - Trimestre X
48. % ARTICLES TRADE MARKETING sobre total ventas - Trimestre X
49. % VENDES SENSE SECCIO sobre total ventas - Trimestre X
50. % CARBURANTS sobre total ventas - Trimestre X
51. % FUNGIBLES INF. DISKETTES sobre total ventas - Trimestre X
52. % GANGAS sobre total ventas - Trimestre X
53. % FUNGIBLES INF. DISKETTES sobre total ventas - Trimestre X
54. CHARCUTERIA
55. CARNE
56. FRUTA
57. PESCADO
58. PANADERIA
59. Tipo tienda
60. Client Talla centro
61. Cierra mediodia
62. Region
63. Provincia
64. Municipio
65. Codi postal
66. Metros cuadrados

Figure B.1: List of features used in version 1.

1. *Tenda*
2. *Max Ticket - Trimestre X*
3. *Mean Ticket - Trimestre X*
4. *Numero medio unidades diarios - Trimestre X*
5. *Numero medio unidades semanales - Trimestre X*
6. *Numero medio unidades mensuales - Trimestre X*
7. *% Unidades del total es venen en Dilluns - Trimestre X*
8. *% Unidades del total es venen en Dimarts - Trimestre X*
9. *% Unidades del total es venen en Dimecres - Trimestre X*
10. *% Unidades del total es venen en Dijous - Trimestre X*
11. *% Unidades del total es venen en Divendres - Trimestre X*
12. *% Unidades del total es venen en Dissabte - Trimestre X*
13. *% Unidades del total es venen en Diumenge - Trimestre X*
14. *Unidades venuts marca client- Trimestre X*
15. *Unidades venuts marca propia - Trimestre X*
16. *Numero referencias - Trimestre X*
17. *Relacio unidades client vs No client- Trimestre X*
18. *Relacio unidades venuts vs numero referencies - Trimestre X*
19. *Numero medio unidades por ticket - Trimestre X*
20. *Numero medio referencias por ticket - Trimestre X*
21. *TOP 1 Referencia aparece en mas tickets - Trimestre X*
22. *TOP 2 Referencia aparece en mas tickets - Trimestre X*
23. *TOP 3 Referencia aparece en mas tickets - Trimestre X*
24. *TOP 1 Referencia se venden mas unidades - Trimestre X*
25. *TOP 2 Referencia se venden mas unidades - Trimestre X*
26. *TOP 3 Referencia se venden mas unidades - Trimestre X*
27. *% ALIMENTACIO SECA sobre total ventas - Trimestre X*
28. *% FLECA I PASTISSERIA sobre total ventas - Trimestre X*
29. *% FORMATGES sobre total ventas - Trimestre X*
30. *% CONSERVES sobre total ventas - Trimestre X*
31. *% DERIVATS LACTIS sobre total ventas - Trimestre X*
32. *% XARCUTERIA TRADICIONAL sobre total ventas - Trimestre X*
33. *% PEIXOS I MARISC sobre total ventas - Trimestre X*
34. *% CARNS sobre total ventas - Trimestre X*
35. *% LIQUIDS I BEGUDES sobre total ventas - Trimestre X*
36. *% LLETS I BATUTS sobre total ventas - Trimestre X*
37. *% DROGUERIA sobre total ventas - Trimestre X*
38. *% BASSAR sobre total ventas - Trimestre X*
39. *% PERFUMERIA sobre total ventas - Trimestre X*
40. *% PLATS CUINATS/REFRIGERATS sobre total ventas - Trimestre X*
41. *% CONGELATS sobre total ventas - Trimestre X*
42. *% FRUITES I HORTALICES sobre total ventas - Trimestre X*
43. *CHARCUTERIA*
44. *CARNICERIA*

45. FRUTERIA
46. PESCADERIA
47. PANADERIA
48. Parking
49. Tipo tienda client
50. Talla centro
51. Cierra mediodia
52. Facturacion trimestral - Trimestre X
53. Venta/m2 - Trimestre X
54. Items 80/20 - Trimestre X
55. Region
56. Provincia
57. Municipio
58. Codi postal
59. Metros cuadrados

Figure B.2: List of features used in version 2.

1. *Tenda*
2. *Max Ticket - Trimestre X*
3. *Mean Ticket - Trimestre X*
4. *Numero medio unidades diarios - Trimestre X*
5. *Numero medio unidades semanales - Trimestre X*
6. *Numero medio unidades mensuales - Trimestre X*
7. *% Unidades del total es venen en Dilluns - Trimestre X*
8. *% Unidades del total es venen en Dimarts - Trimestre X*
9. *% Unidades del total es venen en Dimecres - Trimestre X*
10. *% Unidades del total es venen en Dijous - Trimestre X*
11. *% Unidades del total es venen en Divendres - Trimestre X*
12. *% Unidades del total es venen en Dissabte - Trimestre X*
13. *% Unidades del total es venen en Diumenge - Trimestre X*
14. *Unidades venuts marca client - Trimestre X*
15. *Unidades venuts marca No propia - Trimestre X*
16. *Numero referencias - Trimestre X*
17. *Relacio unidades client vs No client- Trimestre X*
18. *Relacio unidades venuts vs numero referencies - Trimestre X*
19. *Numero medio unidades por ticket - Trimestre X*
20. *Numero medio referencias por ticket - Trimestre X*
21. *TOP 1 Referencia aparece en mas tickets - Trimestre X*
22. *TOP 2 Referencia aparece en mas tickets - Trimestre X*
23. *TOP 3 Referencia aparece en mas tickets - Trimestre X*
24. *TOP 1 Referencia se venden mas unidades - Trimestre X*
25. *TOP 2 Referencia se venden mas unidades - Trimestre X*
26. *TOP 3 Referencia se venden mas unidades - Trimestre X*
27. *% ALIMENTACIO SECA sobre total ventas - Trimestre X*
28. *% FLECA I PASTISSERIA sobre total ventas - Trimestre X*
29. *% FORMATGES sobre total ventas - Trimestre X*
30. *% CONSERVES sobre total ventas - Trimestre X*
31. *% DERIVATS LACTIS sobre total ventas - Trimestre X*
32. *% XARCUTERIA TRADICIONAL sobre total ventas - Trimestre X*
33. *% PEIXOS I MARISC sobre total ventas - Trimestre X*
34. *% CARNS sobre total ventas - Trimestre X*
35. *% LIQUIDS I BEGUDES sobre total ventas - Trimestre X*
36. *% LLETS I BATUTS sobre total ventas - Trimestre X*
37. *% DROGUERIA sobre total ventas - Trimestre X*
38. *% BASSAR sobre total ventas - Trimestre X*
39. *% PERFUMERIA sobre total ventas - Trimestre X*
40. *% PLATS CUINATS/REFRIGERATS sobre total ventas - Trimestre X*
41. *% CONGELATS sobre total ventas - Trimestre X*
42. *% FRUITES I HORTALICES sobre total ventas - Trimestre X*
43. *% ALIMENTACIO SECA Participacio - Trimestre X*
44. *% FLECA I PASTISSERIA Participacio - Trimestre X*

45. % FORMATGES Participacio - Trimestre X
46. % CONSERVES Participacio - Trimestre X
47. % DERIVATS LACTIS Participacio - Trimestre X
48. % XARCUTERIA TRADICIONAL Participacio - Trimestre X
49. % PEIXOS I MARISC Participacio - Trimestre X
50. % CARNS Participacio - Trimestre X
51. % LIQUIDS I BEGUDES Participacio - Trimestre X
52. % LLETS I BATUTS Participacio - Trimestre X
53. % DROGUERIA Participacio - Trimestre X
54. % BASSAR Participacio - Trimestre X
55. % PERFUMERIA Participacio - Trimestre X
56. % PLATS CUINATS/REFRIGERATS Participacio - Trimestre X
57. CONGELATS Participacio - Trimestre X
58. % FRUITES I HORTALICES Participacio - Trimestre X
59. % ALIMENTACIO SECA Penetracio - Trimestre X
60. % FLECA I PASTISSERIA Penetracio - Trimestre X
61. % FORMATGES Penetracio - Trimestre X
62. % CONSERVES Penetracio - Trimestre X
63. % DERIVATS LACTIS Penetracio - Trimestre X
64. % XARCUTERIA TRADICIONAL Penetracio - Trimestre X
65. % PEIXOS I MARISC Penetracio - Trimestre X
66. % CARNS Penetracio - Trimestre X
67. % LIQUIDS I BEGUDES Penetracio - Trimestre X
68. % LLETS I BATUTS Penetracio - Trimestre X
69. % DROGUERIA Penetracio - Trimestre X
70. % BASSAR Penetraco - Trimestre X,
71. % PERFUMERIA Penetracio - Trimestre X
72. % PLATS CUINATS/REFRIGERATS Penetracio - Trimestre X
73. % CONGELATS Penetracio - Trimestre X
74. % FRUITES I HORTALICES Penetracio - Trimestre X
75. CHARCUTERIA
76. CARNICERIA
77. 'FRUTERIA
78. PESCADERIA
79. PANADERIA
80. Parking
81. Tipo tienda client
82. Talla centro
83. Cierra mediodia
84. Facturacion trimestral - Trimestre X
85. Venta/m2 - Trimestre X
86. Items 80/20 - Trimestre X
87. Region
88. Provincia

89. Municipio
90. Codi postal
91. Metros cuadrados

Figure B.3: List of features used in version 3.

1. *Tenda*
2. *Mean Ticket - Trimestre X*
3. *Numero medio unidades diarios - Trimestre X*
4. *Numero medio unidades semanales - Trimestre X*
5. *Numero medio unidades mensuales - Trimestre X*
6. *% Unidades del total es venen en Dilluns - Trimestre X*
7. *% Unidades del total es venen en Dimarts - Trimestre X*
8. *% Unidades del total es venen en Dimecres - Trimestre X*
9. *% Unidades del total es venen en Dijous - Trimestre X*
10. *% Unidades del total es venen en Divendres - Trimestre X*
11. *% Unidades del total es venen en Dissabte - Trimestre X*
12. *% Unidades del total es venen en Diumenge - Trimestre X*
13. *Unidades venuts marca client- Trimestre X*
14. *Unidades venuts marca propia - Trimestre X*
15. *Numero referencias - Trimestre X*
16. *Relacio unidades client vs No client- Trimestre X*
17. *Relacio unidades venuts vs numero referencies - Trimestre X*
18. *Numero medio unidades por ticket - Trimestre X*
19. *Numero medio referencias por ticket - Trimestre X*
20. *Unidades % ALIMENTACIO SECA sobre total - Trimestre X*
21. *Unidades % FLECA I PASTISSERIA sobre total - Trimestre X*
22. *Unidades % FORMATGES sobre total - Trimestre X*
23. *Unidades % CONSERVES sobre total - Trimestre X*
24. *Unidades % DERIVATS LACTIS sobre total - Trimestre X*
25. *Unidades % XARCUTERIA TRADICIONAL sobre total - Trimestre X*
26. *Unidades % PEIXOS I MARISC sobre total - Trimestre X*
27. *Unidades % CARNS sobre total - Trimestre X*
28. *Unidades % LIQUIDS I BEGUDES sobre total - Trimestre X*
29. *Unidades % LLETS I BATUTS sobre total - Trimestre X*
30. *Unidades % DROGUERIA sobre total - Trimestre X*
31. *Unidades % BASAR sobre total - Trimestre X*
32. *Unidades % PERFUMERIA sobre total - Trimestre X*
33. *Unidades % PLATS CUINATS/REFRIGERATS sobre total - Trimestre X*
34. *Unidades % CONGELATS sobre total - Trimestre X*
35. *Unidades % FRUITES I HORTALICES sobre total - Trimestre X*
36. *Participacio % ALIMENTACIO SECA - Trimestre X*
37. *Participacio % FLECA I PASTISSERIA - Trimestre X*
38. *Participacio % FORMATGES - Trimestre X*
39. *Participacio % CONSERVES - Trimestre X*
40. *Participacio % DERIVATS LACTIS - Trimestre X*
41. *Participacio % XARCUTERIA TRADICIONAL - Trimestre X*
42. *Participacio % PEIXOS I MARISC - Trimestre X*
43. *Participacio % CARNS - Trimestre X*
44. *Participacio % LIQUIDS I BEGUDES - Trimestre X*

45. Participacio % LLETS I BATUTS - Trimestre X
46. Participacio % DROGUERIA - Trimestre X
47. Participacio % BASAR - Trimestre X
48. Participacio % PERFUMERIA - Trimestre X
49. Participacio % PLATS CUINATS/REFRIGERATS - Trimestre X
50. Participacio % CONGELATS - Trimestre X
51. Participacio % FRUITES I HORTALICES - Trimestre X
52. Penetracio % ALIMENTACIO SECA - Trimestre X
53. Penetracio % FLECA I PASTISSERIA - Trimestre X
54. Penetracio % FORMATGES - Trimestre X
55. Penetracio % CONSERVES - Trimestre X
56. Penetracio % DERIVATS LACTIS - Trimestre X
57. Penetracio % XARCUTERIA TRADICIONAL - Trimestre X
58. Penetracio % PEIXOS I MARISC - Trimestre X
59. Penetracio % CARNS - Trimestre X
60. Penetracio % LIQUIDS I BEGUDES - Trimestre X
61. Penetracio % LLETS I BATUTS - Trimestre X
62. Penetracio % DROGUERIA - Trimestre X
63. Penetracio % BASAR - Trimestre X
64. Penetracio % PERFUMERIA - Trimestre X
65. Penetracio % PLATS CUINATS/REFRIGERATS - Trimestre X
66. Penetracio % CONGELATS - Trimestre X
67. Penetracio % FRUITES I HORTALICES - Trimestre X
68. CHARCUTERIA
69. CARNICERIA
70. FRUTERIA
71. PESCADERIA
72. PANADERIA
73. Parking
74. Ensena
75. Cierra mediodia
76. Facturacion trimestral - Trimestre X
77. Venta/m2 - Trimestre X
78. Items 80/20 - Trimestre X
79. Region
80. Provincia
81. Metros cuadrados

Figure B.4: List of features used in version 4.

1. Tenda
2. Mean Ticket - Trimestre X
3. Numero medio unidades diarios - Trimestre X
4. Numero medio unidades semanales - Trimestre X
5. Numero medio unidades mensuales - Trimestre X
6. % Unidades del total es venen en Dilluns - Trimestre X
7. % Unidades del total es venen en Dimarts - Trimestre X
8. % Unidades del total es venen en Dimecres - Trimestre X
9. % Unidades del total es venen en Dijous - Trimestre X
10. % Unidades del total es venen en Divendres - Trimestre X
11. % Unidades del total es venen en Dissabte - Trimestre X
12. % Unidades del total es venen en Diumenge - Trimestre X
13. Unidades venuts marca client- Trimestre X
14. Unidades venuts marca propia - Trimestre X
15. Numero referencias - Trimestre X
16. Relacio unidades client vs No client- Trimestre X
17. Relacio unidades venuts vs numero referencies - Trimestre X
18. Numero medio unidades por ticket - Trimestre X
19. Numero medio referencias por ticket - Trimestre X
20. Unidades % ALIMENTACIO SECA - Trimestre X
21. Unidades % FLECA I PASTISSERIA - Trimestre X
22. Unidades % FORMATGES - Trimestre X
23. Unidades % CONSERVES - Trimestre X
24. Unidades % DERIVATS LACTIS - Trimestre X
25. Unidades % XARCUTERIA TRADICIONAL - Trimestre X
26. Unidades % PEIXOS I MARISC - Trimestre X
27. Unidades % CARNS - Trimestre X
28. Unidades % LIQUIDS I BEGUDES - Trimestre X
29. Unidades % LLETS I BATUTS - Trimestre X
30. Unidades % DROGUERIA - Trimestre X
31. Unidades % BASAR - Trimestre X
32. Unidades % PERFUMERIA - Trimestre X
33. Unidades % PLATS CUINATS/REFRIGERATS - Trimestre X
34. Unidades % CONGELATS - Trimestre X
35. Unidades % FRUITES I HORTALICES - Trimestre X
36. Participacio % ALIMENTACIO SECA - Trimestre X
37. Participacio % FLECA I PASTISSERIA - Trimestre X
38. Participacio % FORMATGES - Trimestre X
39. Participacio % CONSERVES - Trimestre X
40. Participacio % DERIVATS LACTIS - Trimestre X
41. Participacio % XARCUTERIA TRADICIONAL - Trimestre X
42. Participacio % PEIXOS I MARISC - Trimestre X
43. Participacio % CARNS - Trimestre X
44. Participacio % LIQUIDS I BEGUDES - Trimestre X

45. Participacio % LLETS I BATUTS - Trimestre X
46. Participacio % DROGUERIA - Trimestre X
47. Participacio % BASAR - Trimestre X
48. Participacio % PERFUMERIA - Trimestre X
49. Participacio % PLATS CUINATS/REFRIGERATS - Trimestre X
50. Participacio % CONGELATS - Trimestre X
51. Participacio % FRUITES I HORTALICES - Trimestre X
52. CHARCUTERIA
53. CARNICERIA
54. FRUTERIA
55. PESCADERIA
56. PANADERIA
57. Parking
58. Ensena
59. Cierra mediodia
60. Facturacion trimestral - Trimestre 1
61. Venta/m2 - Trimestre 1
62. Region
63. Provincia
64. Metros cuadrados
65. Coverage % Families

Figure B.5: List of features used in version 5.